

# Retrospective clinical data harmonisation reporting using R and Quarto

---

Jeremy Selva 

@JauntyJJS    

<https://jeremy-selva.netlify.app> 

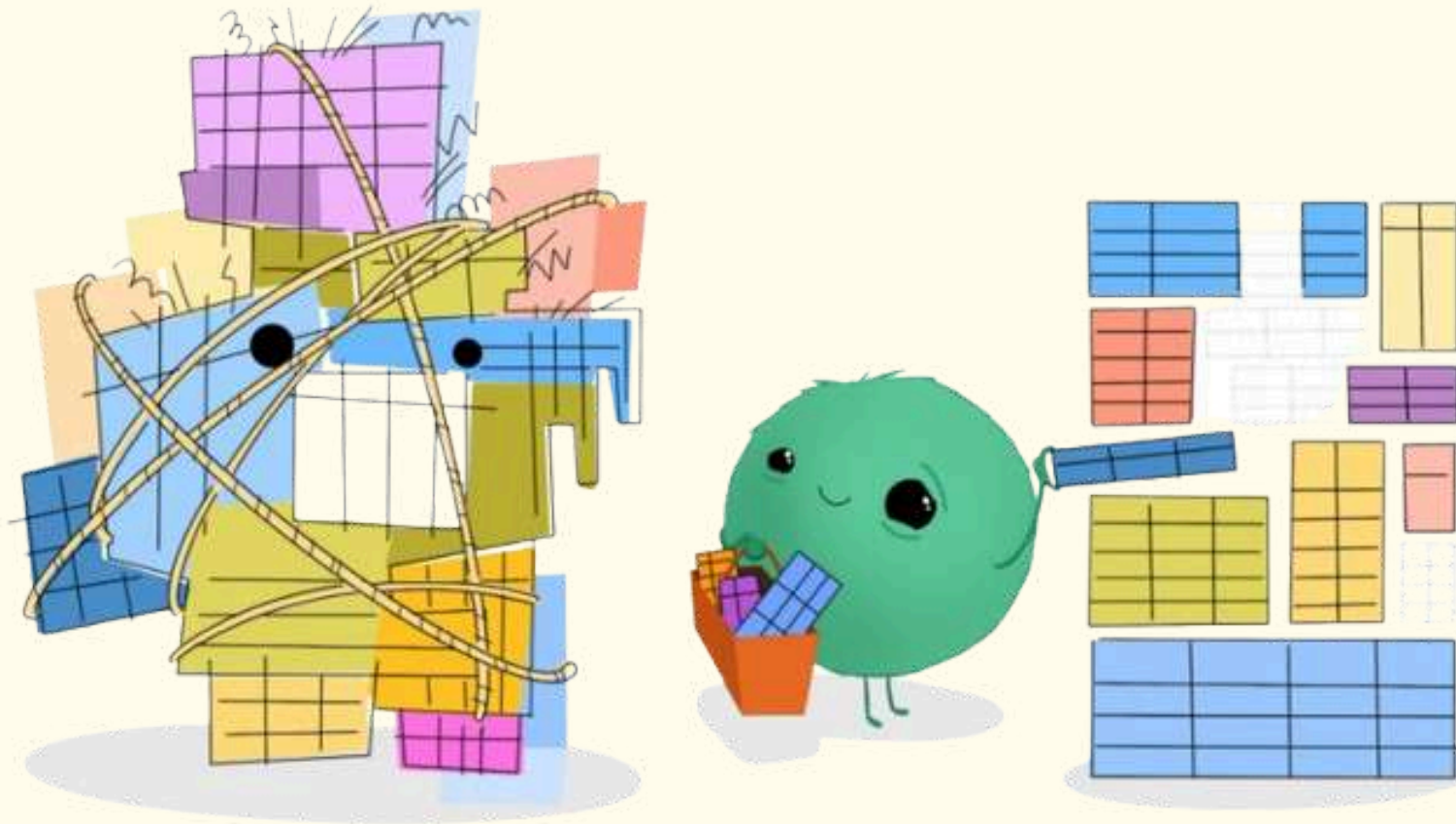
For useR! 2025 

9 August 2025



# whoami

Research Officer from [National Heart Centre Singapore](#) who collects, cleans and harmonises clinical data.



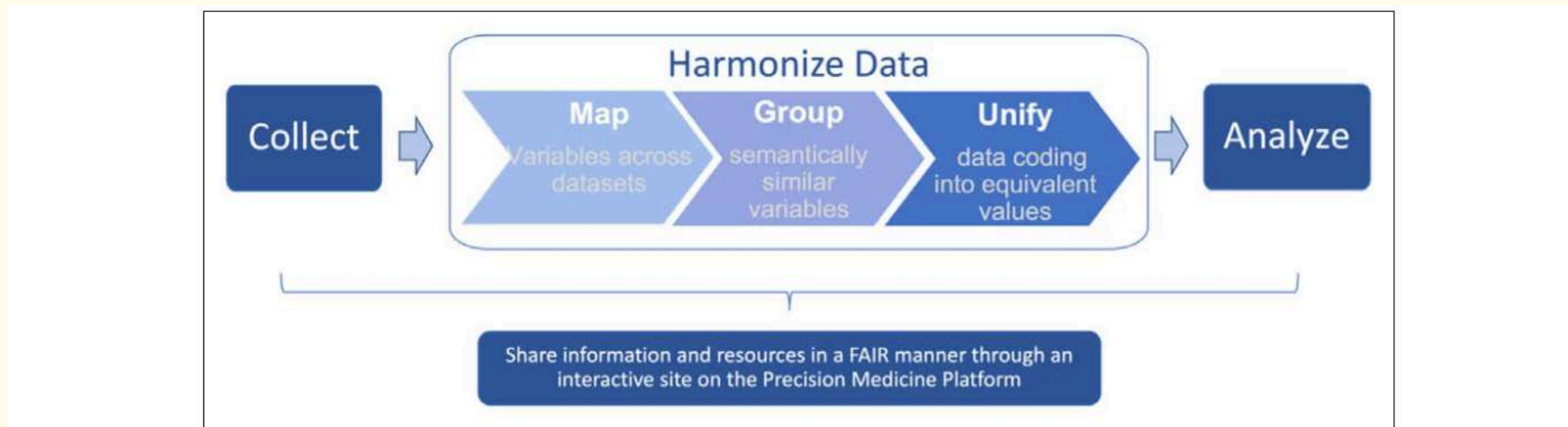
Taming the Data Beast from “[Cleaning Medical Data with R](#)” workshop by Shannon Pileggi, Crystal Lewis and Peter Higgins presented at R/Medicine 2023.

Illustrated by [Allison Horst](#).

# About Data Harmonisation

Data harmonisation is part of data wrangling process where

- Similar variables from different datasets are identified.
- Grouped based on a generalised concept they represent.
- Transformed into unified harmonised variables for analysis.



**Figure 1. The data harmonization process.**

Study data variables collected from different sources need to be mapped to one another (step 1), classified into the generalized concepts they represent (step 2), and transformed into unified harmonized variables (step 3) for analysis.

# How it started



Welcome to the project.

Here is the data dictionary and input template file you can use to send the data.

20250310\_data\_input.xlsx - Excel

	B	C	D	E	F	G	H	I	J
1	cohort_unique_id	age_years	sex	height_cm	weight_kg	bsa_m2	bmi	smoke_current	smoke_past
2	patient_01	50	0	167	58	1.64	20.8	1	0
3	patient_02	40	1	185	75	1.96	21.9	0	0

20250310\_data\_dictionary.xlsx - Excel

A	B	C	D
variable_name	variable_label	value_type	value_label
cohort_unique_id	Unique registry ID of the cohort from collaborator.	character	
age_years	Calculated age at time of scan.	positive integer	
sex	Patient's sex at time of scan.	categorical	0 = female, 1 = male, -1 = unknown
height_cm	Patient's height in cm at time of scan.	positive numeric in two decimal places	
weight_kg	Patient's weight in kg at time of scan.	positive numeric in two decimal places	
bsa_m2	Patient's body surface area in m <sup>2</sup> at time of scan.	positive numeric in two decimal places	
bmi	Patient's body mass index at time of scan.	positive numeric in two decimal places	



# How it started



Received with thanks.

We don't have an analyst to do the mapping.

We can do it ourselves but our workload allow us to work on one data field per day...





# How it started

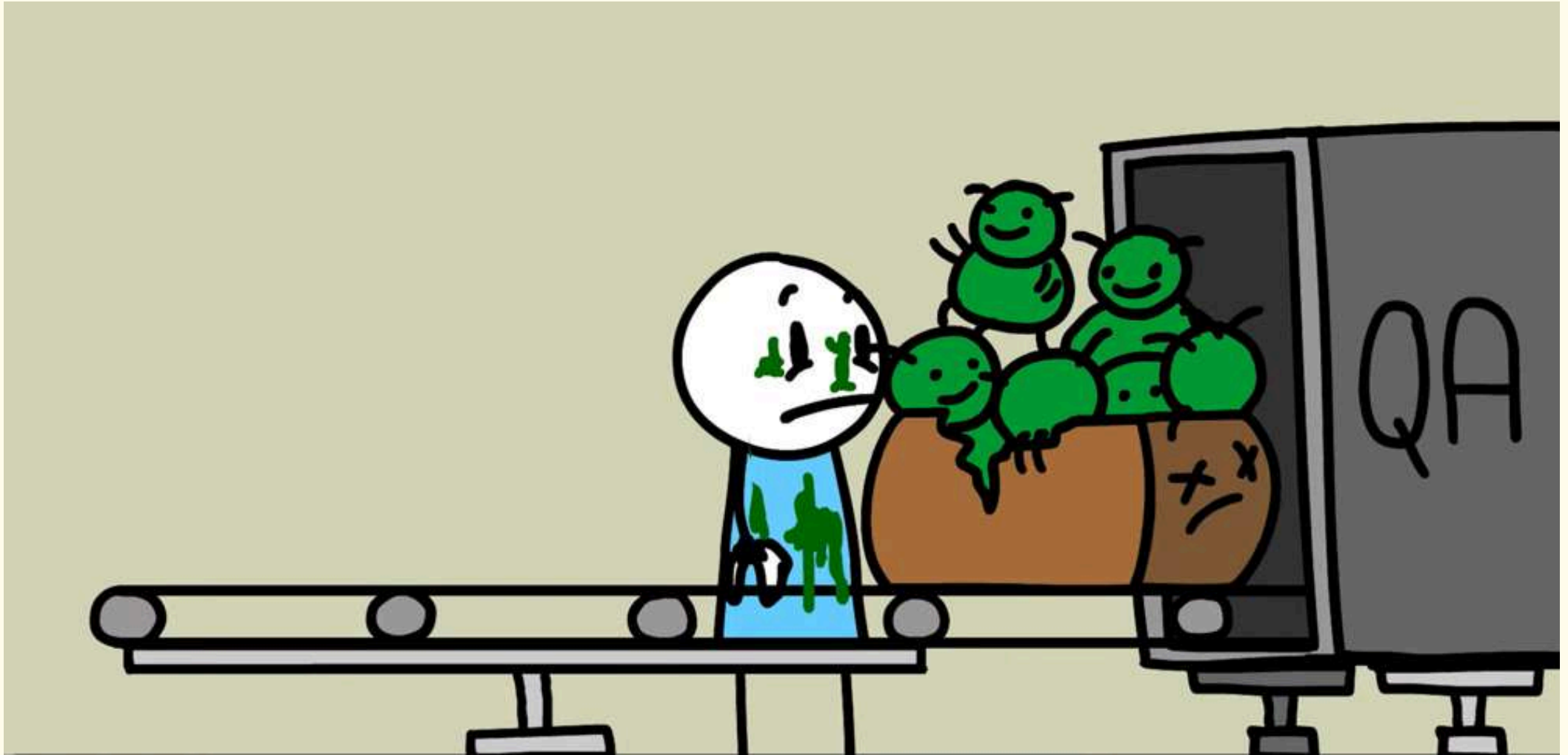


This means it will take at least one year for us to finish (given ~350 data fields to do).

Could we send the data and you do it for us instead ?

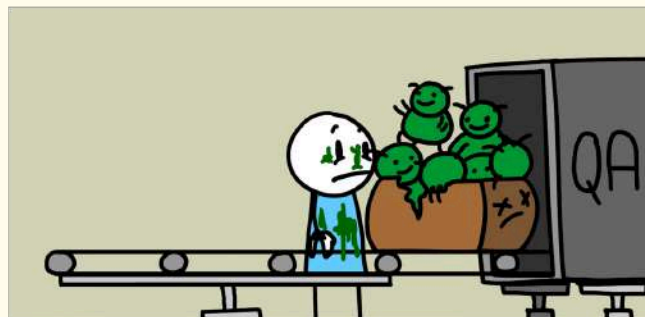
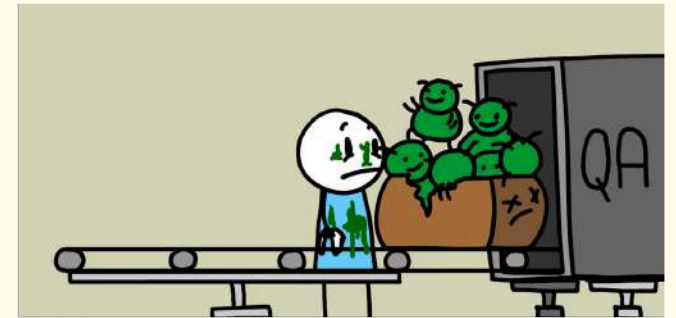
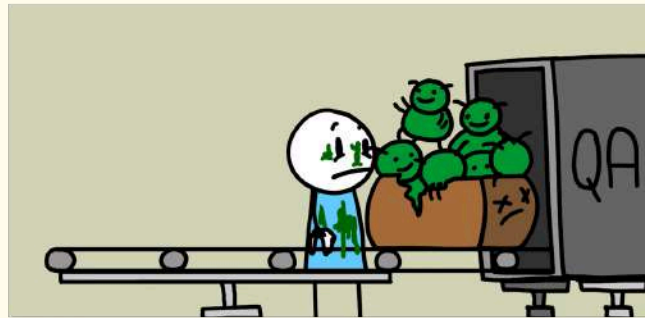
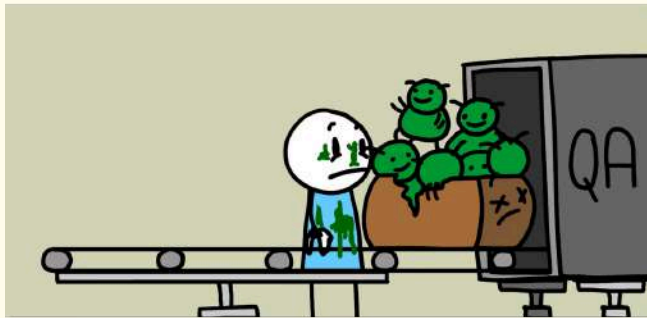
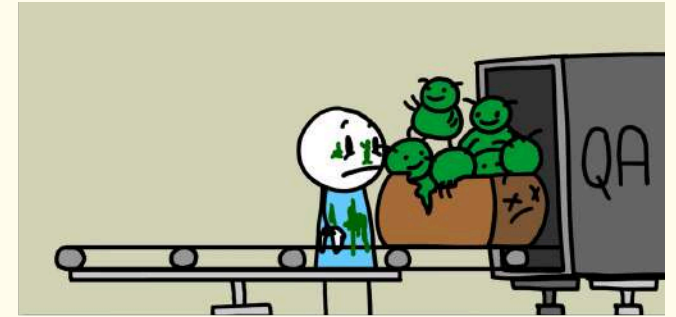
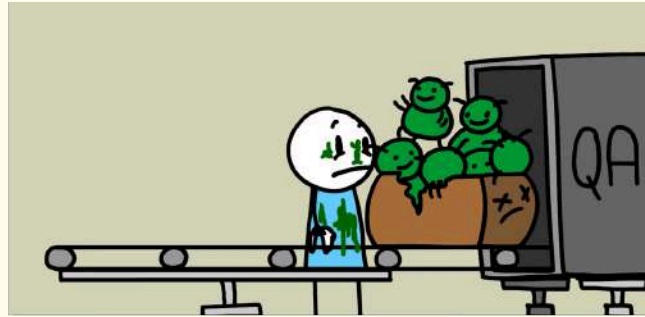
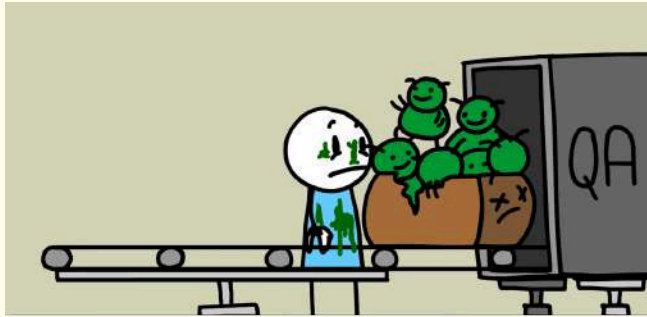


# How it started



snapshot from Ready for QA | MonkeyUser 2SP Animation Video from MonkeyUser.com.

# How it started



snapshot from Ready for QA | MonkeyUser 2SP Animation Video from [MonkeyUser.com](https://monkeyuser.com).



## How it started

Turn my sorrow into opportunities.

# Tackling Formatted Tabular Data from Excel



*10th July 2024*

Jeremy Selva 

@JauntyJJS  

<https://jeremy-selva.netlify.app> 

# How it started



Could you also send the harmonised data back to us with a report on how it is done ?

Our higher management needs it for an audit to show that the data is reliable.



# Additional Motivation

Some data fields just cannot be planned in advanced.

Cohort 1 Race/Ethnicity
Chinese
Indian
Malay
Eurasian
Others

Cohort 2 Race/Ethnicity
White
Black
Asian
Mixed
Others

Cohort 3 Race/Ethnicity	
Race	Ethnicity
White	Hispanic/Latino
Black	Not Hispanic/Latino
Asian	
Native American	
Pacific Islanders	
Others	

Cohort 4 Race/Ethnicity in text	
Latino	Brazilian
White	British
Asian	Chinese
Middle Eastern	Egyptian
Asian	Korean
Asian	Filipino
White	German
Asian	Japanese
Asian	Korean
Asian	Pakistani
Latino	Peruvian
Middle Eastern	Saudi Arabian
African	Ugandan
etc ...	

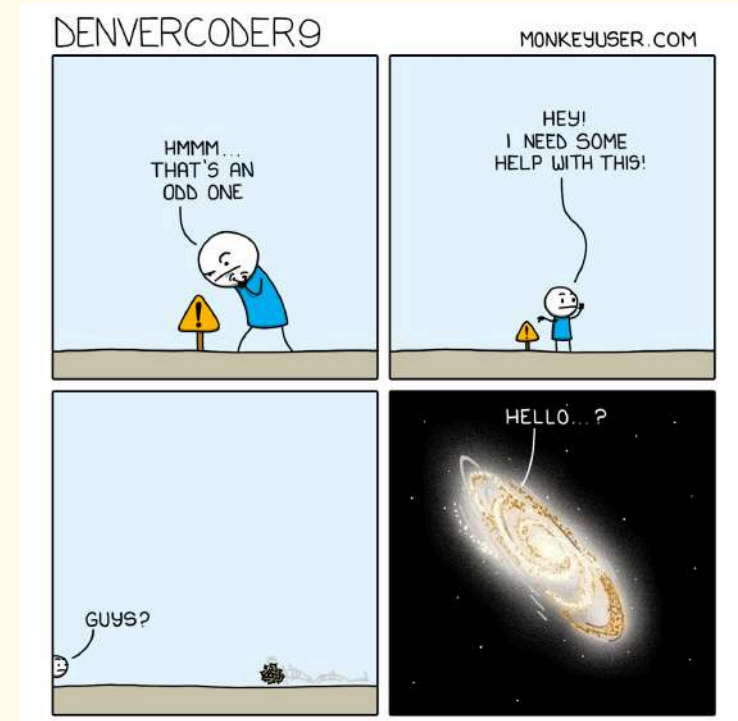
New Cohort Race/Ethnicity
White
African
Southeast Asian
East Asian
South Asian
Other Asians
Middle Eastern
Torres Stratis Islanders
Aboriginal
Others

# Issues

While there are R packages to facilitate data harmonisation,

- **retroharmonize** for survey data.
- **Rmonize** for epidemiological data.
- **psHarmonize** for health and education data.

There are limited resources on how to make a data harmonisation report.



DenverCoder9 from MonkeyUser.com



# Harmonisation Project Template

A template to offer a systematic way to report data harmonisation processes.

[Link: <https://jauntyjjs.github.io/harmonisation/>]

harmonisation 1.0.0.0 Reference

Search for



## Data Harmonisation Project Template

---

### Table of Content

- [Motivation](#)
- [Acknowledgement](#)
- [File Structure](#)
- [Software Installation](#)
- [R Package Installation](#)
- [Using `renv`](#)
- [R Functions Management](#)
- [R Packages Used](#)

### Links

[Browse source code](#)

[Report a bug](#)

### License

[Full license](#)

[MIT](#) + file [LICENSE](#)

### Citation

[Citing harmonisation](#)

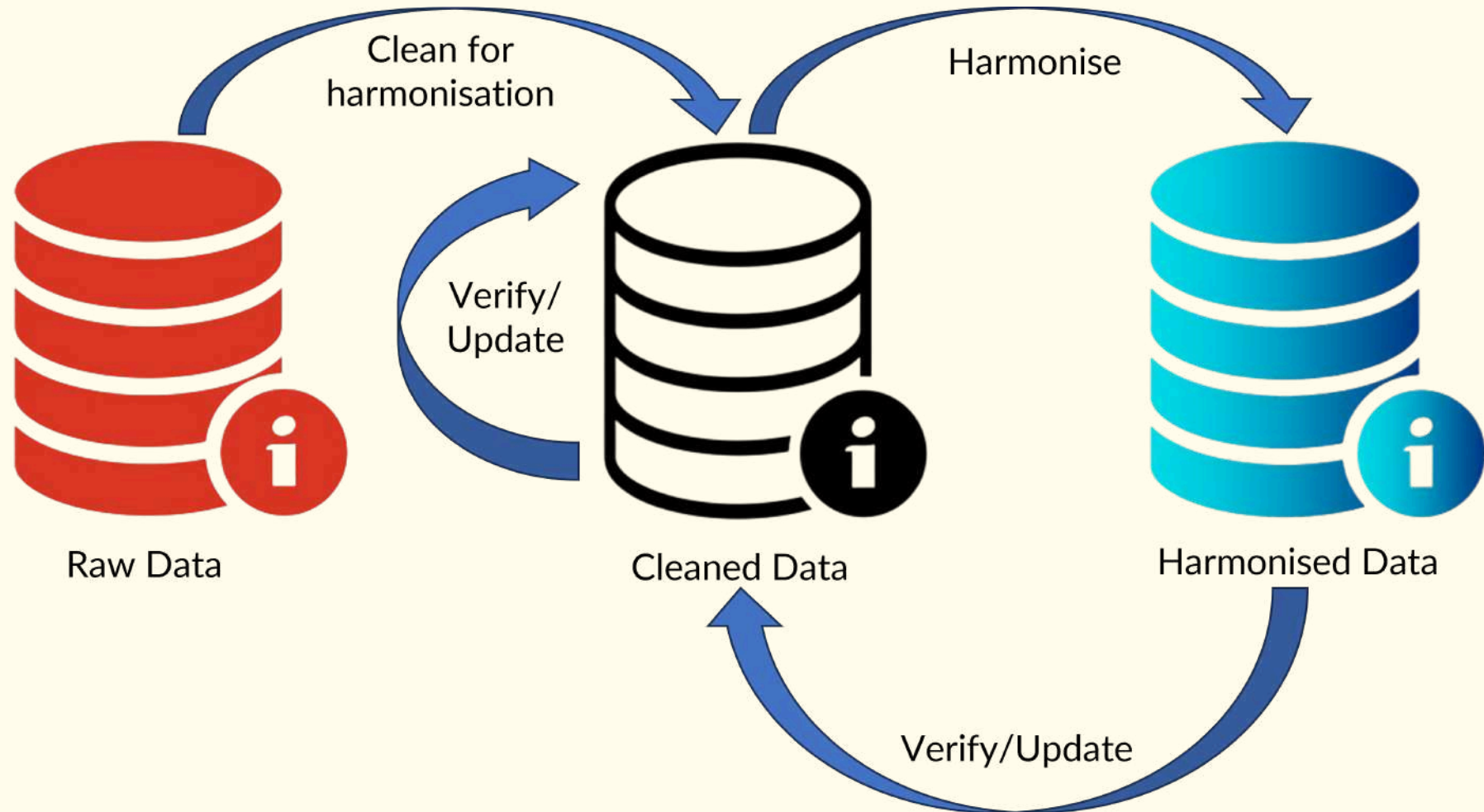
### Developers

Jeremy Selva

Author, maintainer 

# Workflow with collaborators

Collaborator can send the raw data once and you keep updating the cleaned data for harmonisation.



# Workflow with collaborators



Thank you for the harmonised data.

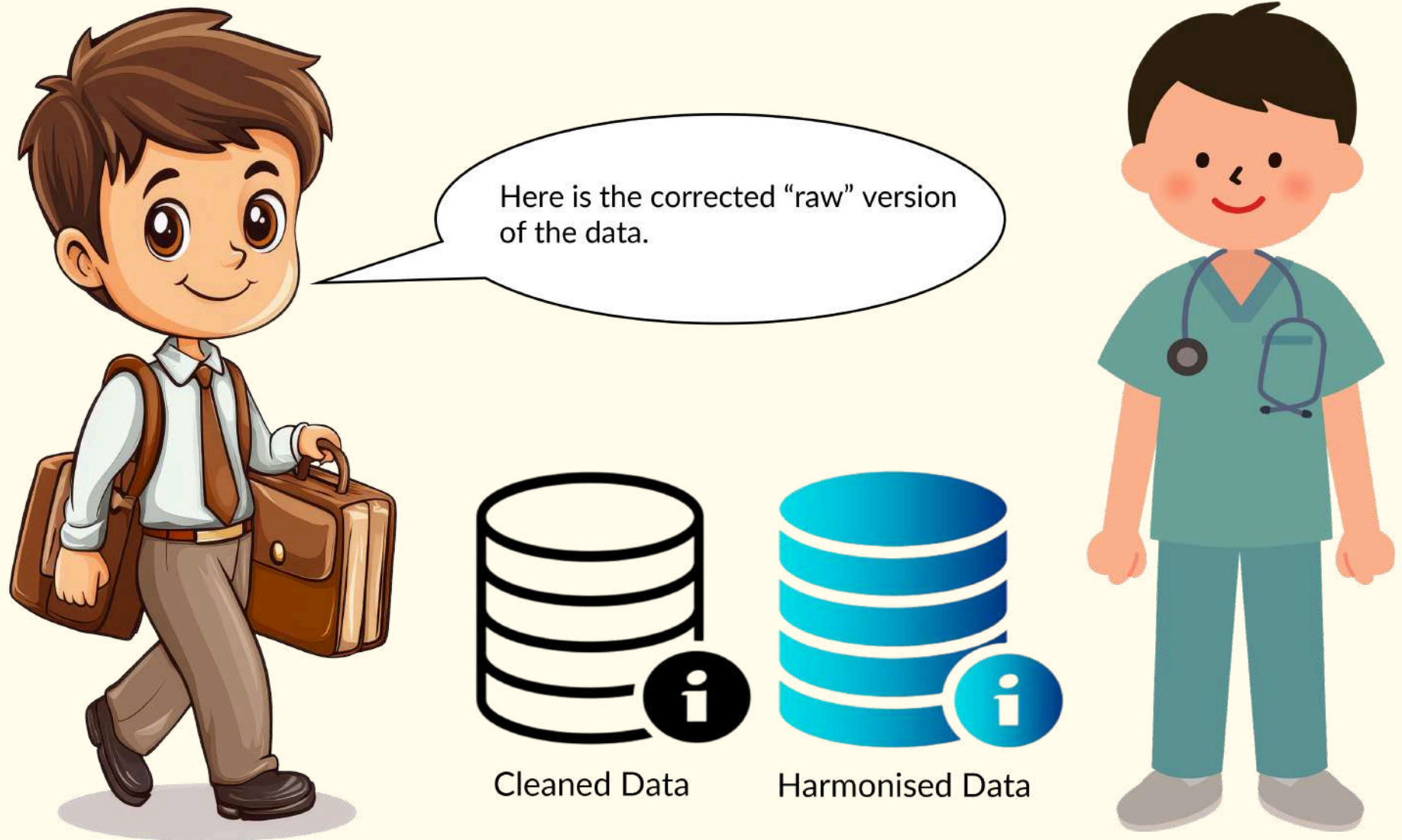
Could we also have the corrected  
“raw” version of the data as well ?



Harmonised Data



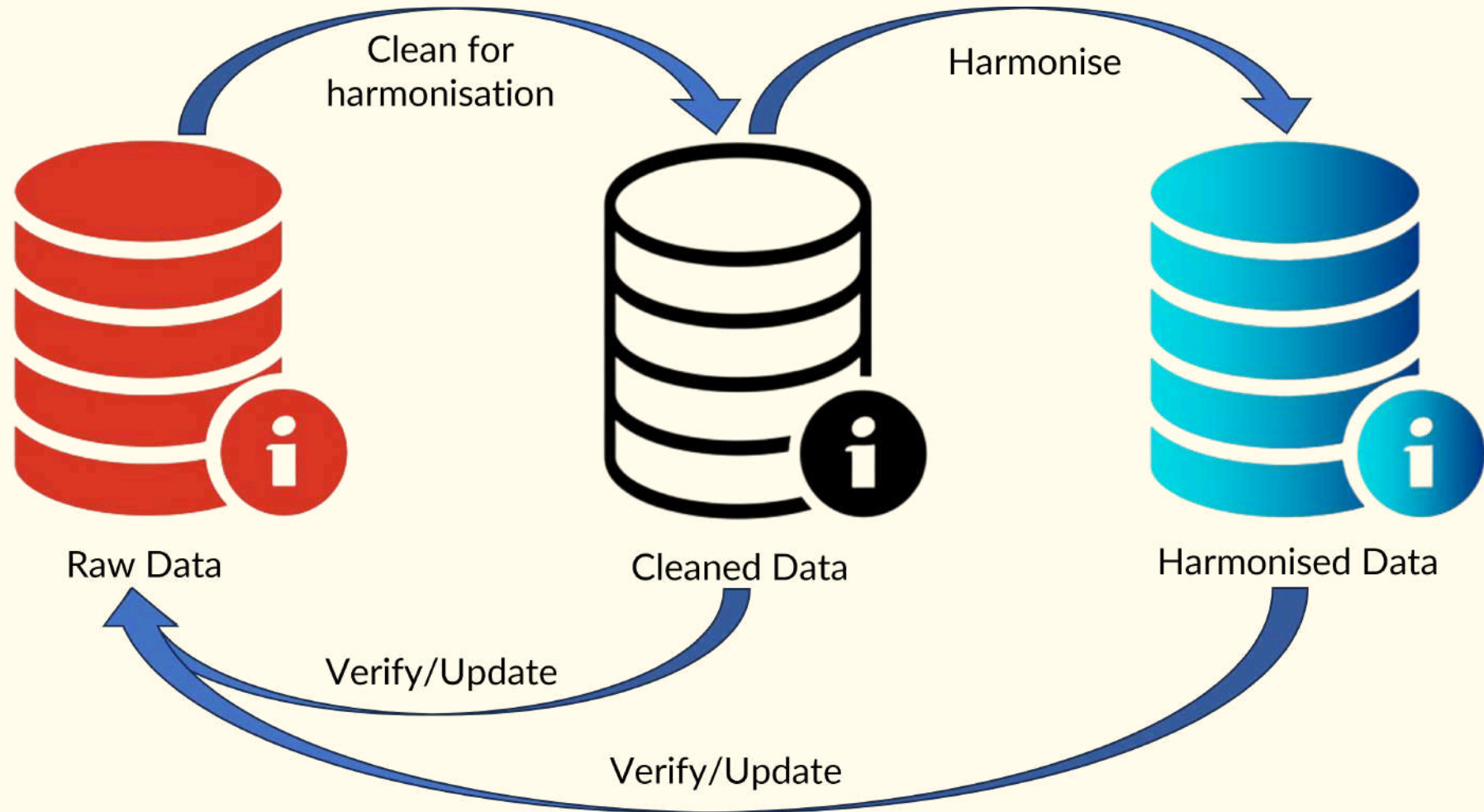
# Workflow with collaborators





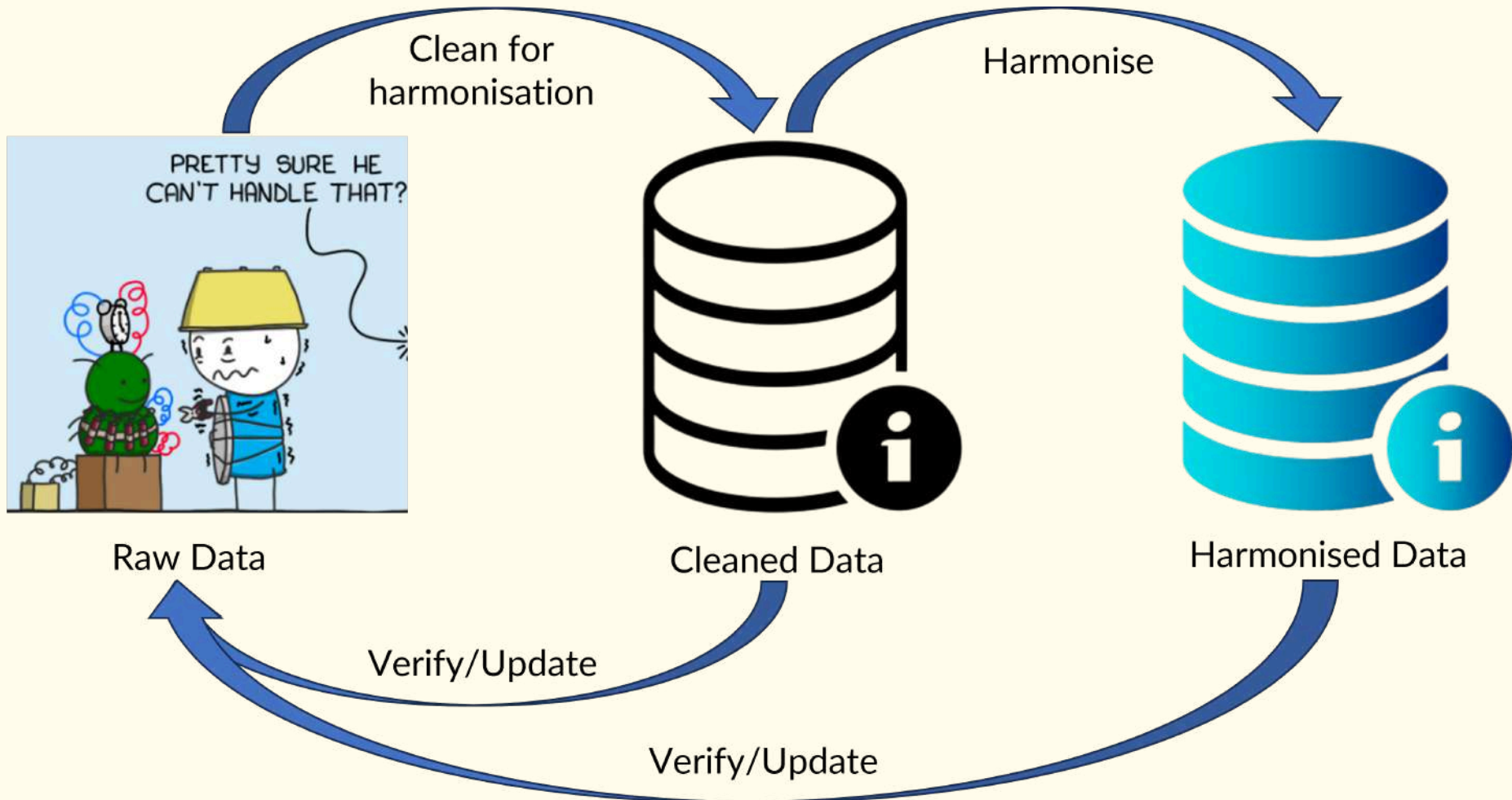
# Workflow with collaborators

Collaborator can update the raw data. For example, adding new clinical data, add more patients, correct errors.



# Workflow with collaborators

New version means new bugs or reopen issues to fix. Is there an automated way to catch warnings/issues when reading these updated files ?



# Automated capture of warnings (csv)

Is there an automated way to catch warnings/issues when reading csv files ?

```
1 cohort_data_csv <- vroom::vroom(  
2   file = here::here("data-raw", "Cohort_csv",  
3     "data_to_harmonise_age_issue.csv"),  
4   delim = ",",  
5   col_select = 1:2,  
6   show_col_types = FALSE,  
7   col_types = list(  
8     ID = vroom::col_character(),  
9     Age = vroom::col_integer()  
10  )  
11 )  
12  
13 head(cohort_data_csv, n = 3)
```

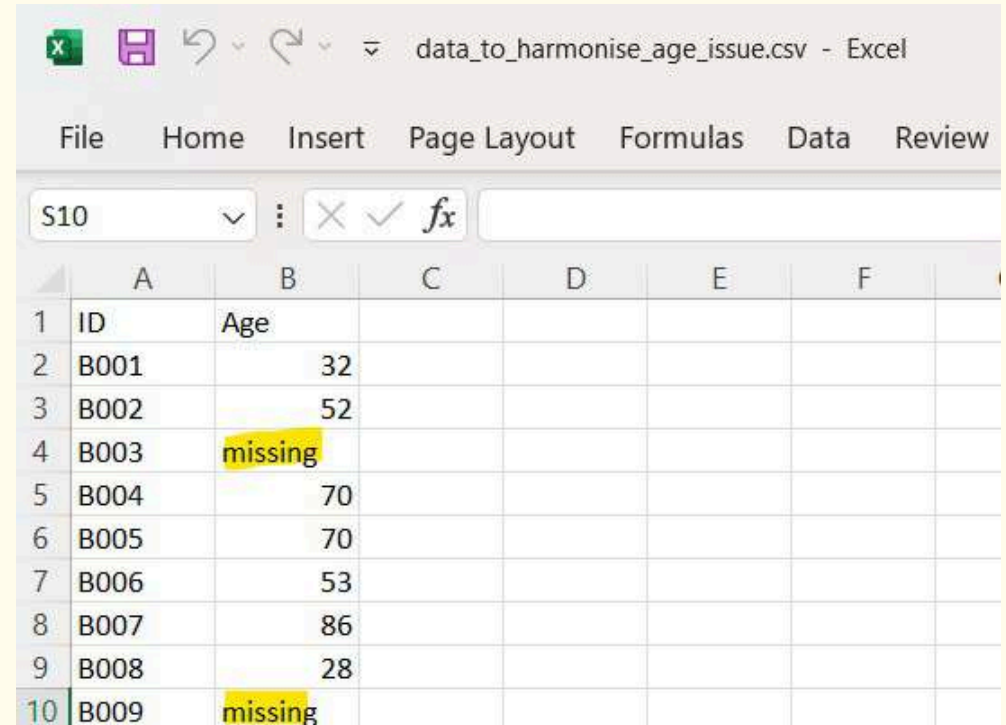
# A tibble: 3 × 2

Warning: One or more parsing issues, call `problems()` on your data frame for details,

e.g.:

```
dat <- vroom(...)  
problems(dat)
```

```
  ID      Age  
  <chr> <int>  
1 B001    32  
2 B002    52  
3 B003    NA
```



The screenshot shows the Microsoft Excel interface with the file "data\_to\_harmonise\_age\_issue.csv" open. The worksheet displays a table with two columns: "ID" and "Age". The data rows are as follows:

	A	B	C	D	E	F
1	ID	Age				
2	B001	32				
3	B002	52				
4	B003	missing				
5	B004	70				
6	B005	70				
7	B006	53				
8	B007	86				
9	B008	28				
10	B009	missing				

The cells containing "missing" (B4 and B10) are highlighted in yellow, indicating parsing issues.

# Automated capture of warnings (csv)

If there are issues with the data, the output of `vroom::problems` will be a tibble.

```
1 cohort_data_csv |>
2   vroom::problems()
```

```
# A tibble: 4 × 5
   row  col expected  actual file
<int> <int> <chr>      <chr> <chr>
1     2     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/use...
2     4     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/use...
3    10     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/use...
4    17     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/use...
```

To check for this automatically, we can use `pointblank::expect_row_count_match`.

```
1 cohort_data_csv |>
2   vroom::problems() |>
3   pointblank::expect_row_count_match(count = 0)
```

```
Error: Row counts for the two tables did not match.
The `expect_row_count_match()` validation failed beyond the absolute threshold level (1).
* failure level (1) >= failure threshold (1)
```



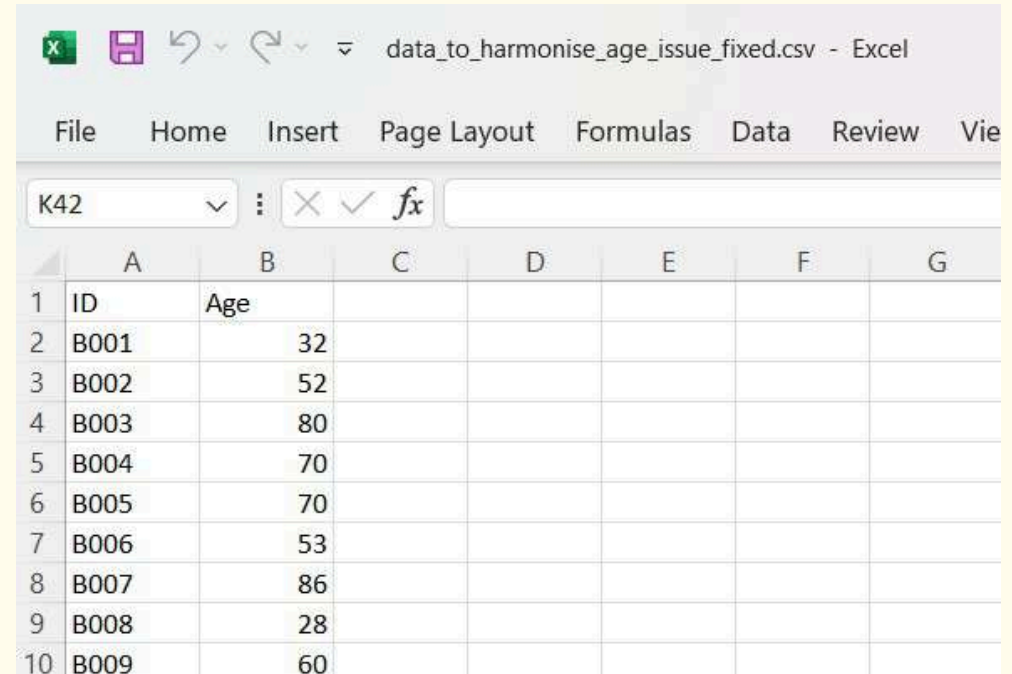
# Automated capture of warnings (csv)

Here is a case with no issues.

```
1 cohort_data_csv <- vroom::vroom(  
2   file = here::here("data-raw", "Cohort_csv",  
3     "data_to_harmonise_age_issue_fixed.csv"),  
4   delim = ",",  
5   col_select = 1:2,  
6   show_col_types = FALSE,  
7   col_types = list(  
8     ID = vroom::col_character(),  
9     Age = vroom::col_integer()  
10  )  
11 )  
12  
13 cohort_data_csv |>  
14 vroom::problems()
```

```
# A tibble: 0 × 5  
# i 5 variables: row <int>, col <int>, expected <chr>, actual  
<chr>, file <chr>
```

```
1 cohort_data_csv |>  
2 vroom::problems() |>  
3 pointblank::expect_row_count_match(count = 0)
```



The screenshot shows the Microsoft Excel interface with the file "data\_to\_harmonise\_age\_issue\_fixed.csv". The spreadsheet has two columns: "ID" and "Age". The data rows are as follows:

	A	B	C	D	E	F	G
1	ID	Age					
2	B001	32					
3	B002	52					
4	B003	80					
5	B004	70					
6	B005	70					
7	B006	53					
8	B007	86					
9	B008	28					
10	B009	60					

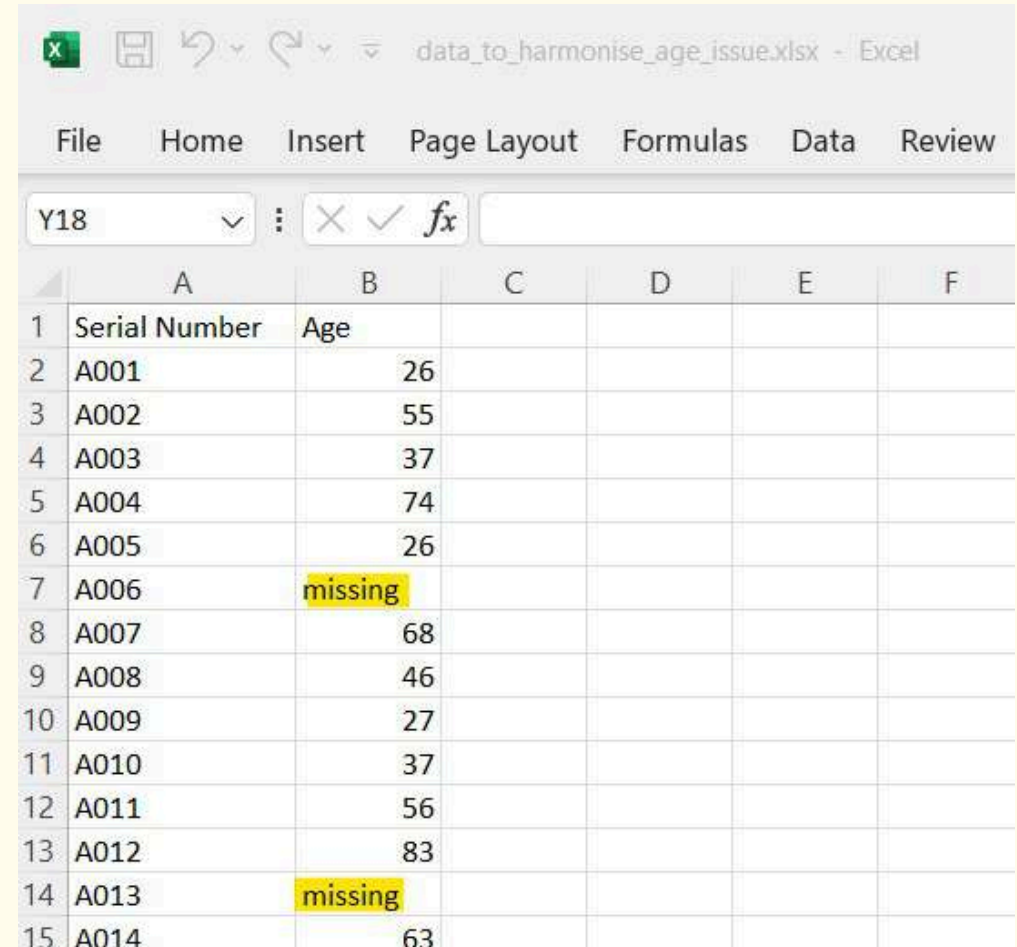
# Automated capture of warnings (Excel)

Is there an automated way to catch warnings/issues when reading Excel files ?

```
1 cohort_data_excel <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c(  
6     "text", "numeric"  
7   )  
8 )
```

Warning: Expecting numeric in B7 / R7C2: got 'missing'

Warning: Expecting numeric in B14 / R14C2: got 'missing'



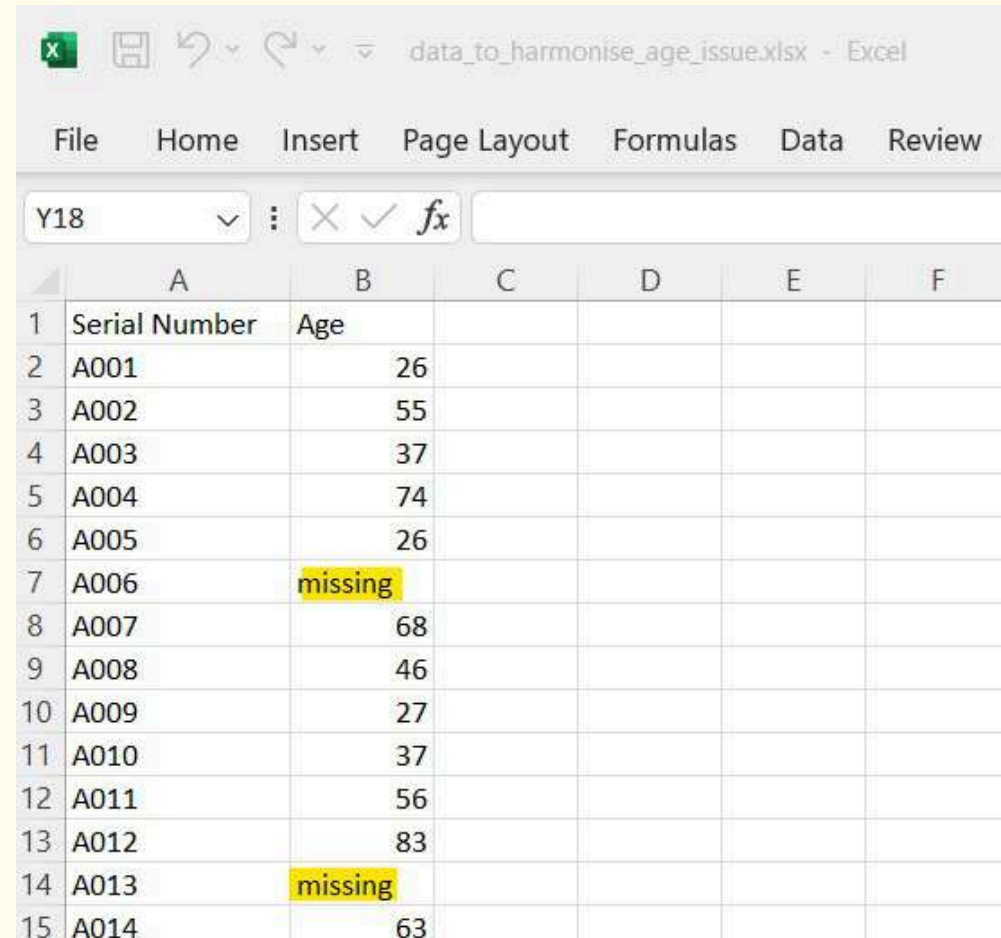
	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	missing				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				
12	A011	56				
13	A012	83				
14	A013	missing				
15	A014	63				

# Automated capture of warnings (Excel)

We can read the Excel file with `testthat::expect_no_condition`.

```
1 testthat::expect_no_condition(  
2   cohort_data_excel <- readxl::read_excel(  
3     path = here::here("data-raw", "Cohort_Excel",  
4       "data_to_harmonise_age_issue.xlsx"),  
5     sheet = "Sheet1",  
6     col_types = c("text", "numeric")  
7   )  
8 )
```

```
Error: Expected `... <- NULL` to run without any conditions.  
i Actually got a <simpleWarning> with text:  
  Expecting numeric in B7 / R7C2: got 'missing'
```



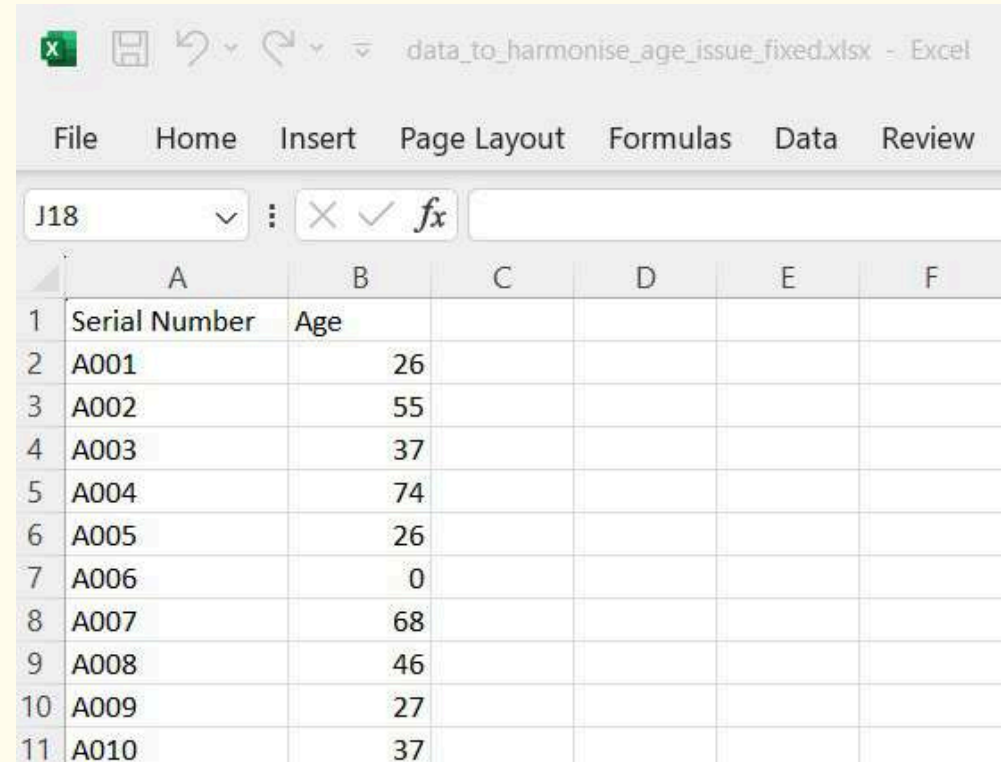
The screenshot shows an Excel spreadsheet titled "data\_to\_harmonise\_age\_issue.xlsx". The spreadsheet has columns A through F and rows 1 through 15. Column A is labeled "Serial Number" and column B is labeled "Age". The data in column B shows ages for various serial numbers, with two instances of "missing" highlighted in yellow: row 7 (A006) and row 14 (A013).

	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	missing				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				
12	A011	56				
13	A012	83				
14	A013	missing				
15	A014	63				

# Automated capture of warnings (Excel)

However, this method means that you will lose the pipe workflow.

```
1 testthat::expect_no_condition(  
2   cohort_data_excel <- readxl::read_excel(  
3     path = here::here("data-raw", "Cohort_Excel",  
4       "data_to_harmonise_age_issue_fixed.xlsx"),  
5     sheet = "Sheet1",  
6     col_types = c("text", "numeric")  
7   )  
8 )  
9  
10 cohort_data_excel <- cohort_data_excel |>  
11   # Check if Serial Number is unique  
12   pointblank::rows_distinct(  
13     columns = "Serial Number",  
14   )
```



	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	0				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				



# Automated capture of warnings (Excel)

We can use the tee pipe operator `%T>%` from `magrittr`.

## With Issues

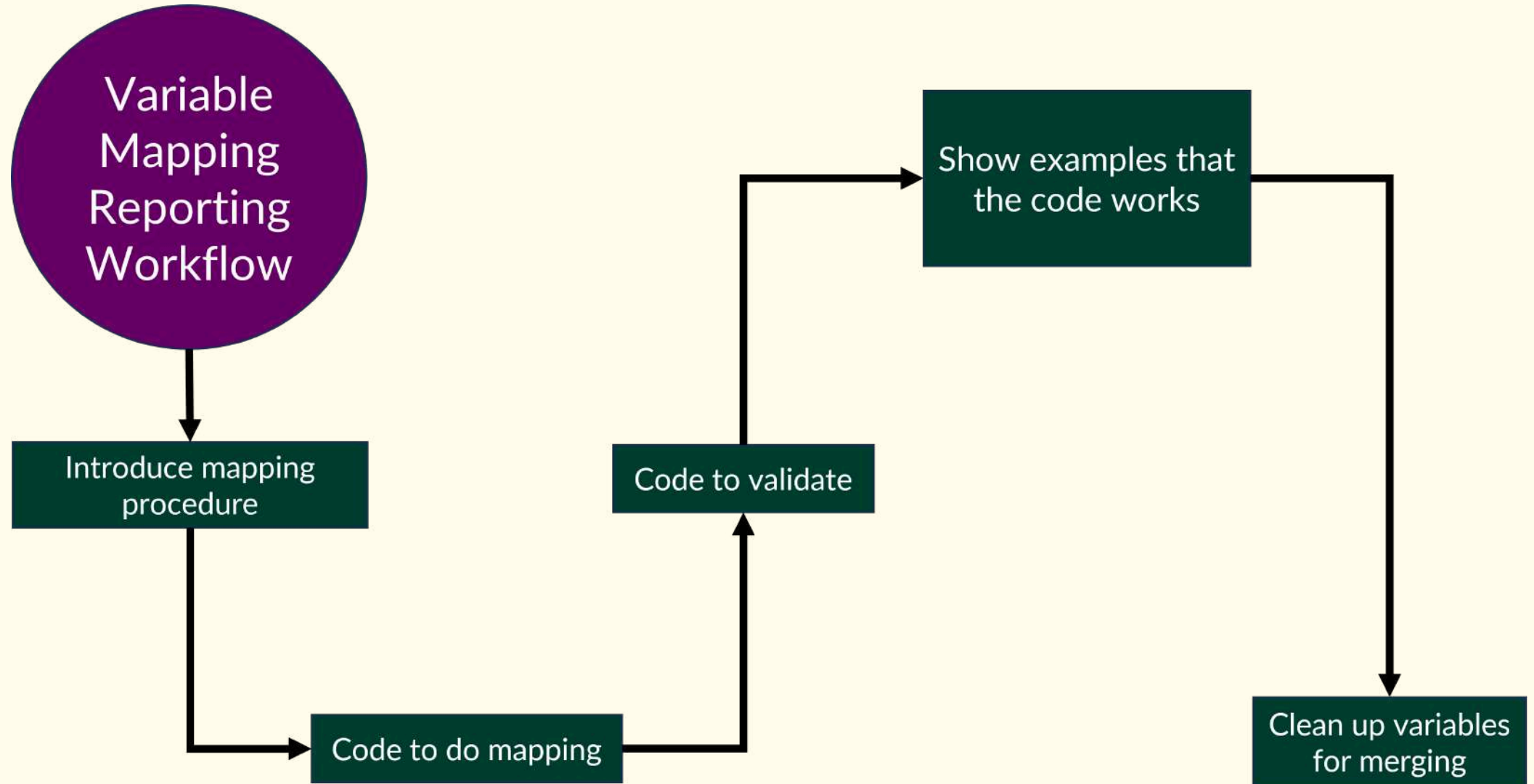
```
1 cohort_data_excel <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c(  
6     "text", "numeric"  
7   )  
8 ) %T>%  
9 testthat::expect_no_condition()
```

Error: Expected `` to run without any conditions.  
i Actually got a <simpleWarning> with text:  
Expecting numeric in B7 / R7C2: got 'missing'

## No Issues

```
1 cohort_data_excel_2 <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue_fixed.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c("text", "numeric")  
6 ) %T>%  
7 testthat::expect_no_condition() |>  
8 # Check if Serial Number is unique  
9 pointblank::rows_distinct(  
10   columns = "Serial Number",  
11 )
```

# Variable Mapping



# Variable Mapping

Let take this data set as an example.

```
1 cohort_csv_data <- vroom::vroom(  
2   file = here::here("data-raw",  
3     "Cohort_csv",  
4     "data_to_harmonise.csv"),  
5   delim = ",",  
6   col_select = 1:8,  
7   show_col_types = FALSE,  
8   col_types = list(  
9     ID = vroom::col_character(),  
10    Age = vroom::col_integer(),  
11    Sex = vroom::col_character(),  
12    Height = vroom::col_double(),  
13    Weight = vroom::col_double(),  
14    `Smoke History` = vroom::col_character(),  
15    `Chest Pain Character` = vroom::col_character(),  
16    Dyspnea = vroom::col_character()  
17  )  
18 ) |>  
19 dplyr::rename(cohort_unique_id = "ID") |>  
20 # Remove rows when the ID value is NA  
21 dplyr::filter(!is.na(.data[["cohort_unique_id"]])) |>  
22 # Remove white spaces in column names
```

cohort_uniq ue_id	Age	Sex	Height	Weight	Sm H
<input type="text"/>	<input type="text"/>	All <input type="button" value="v"/>	<input type="text"/>	<input type="text"/>	A
B001	32	Female	170	63	no
B002	52	Female	167	71	cu sr
B003	80	Male	184	77	no
B004	70	Male	160	83	pa
B005	70	Female	155	61	cu sr

1-5 of 20 rows

Previous1of 4Next

 Download as CSV

# Variable Mapping

Let the reader know how the collaborator's data **Smoke History** is going to be mapped.

## Introduce mapping procedure

```
### Smoking History
```

```
`smoke_current` is grouped as follows:
```

```
```{r}
#| label: smoke current table
#| echo: false
#| message: false
#| warnings: false
#| results: asis
```

```
tabl <- "
```

```
+-----+-----+
| Smoke History | smoke_current |
+=====+=====+
| non-smoker    | 0             |
+-----+-----+
| past smoker   | 0             |
+-----+-----+
| current smoker| 1             |
+-----+-----+
| NA            | -1            |
+-----+-----+
```

```
"
cat(tabl)
```
```

## 2.3 Smoking History

**smoke\_current** is grouped as follows:

| Smoke History  | smoke_current |
|----------------|---------------|
| non-smoker     | 0             |
| past smoker    | 0             |
| current smoker | 1             |
| NA             | -1            |

**smoke\_past** is grouped as follows:

| Smoke History  | smoke_past |
|----------------|------------|
| non-smoker     | 0          |
| past smoker    | 1          |
| current smoker | 0          |
| NA             | -1         |

We do a check to ensure that we can only have these scenarios

- **smoke\_current** as 1 and **smoke\_past** as 0 for current smokers
- **smoke\_current** as 0 and **smoke\_past** as 1 for past smokers
- **smoke\_current** as 0 and **smoke\_past** as 0 for non-smokers
- **smoke\_current** as -1 and **smoke\_past** as -1 for unknown

# Variable Mapping

Code to do mapping

```
1 smoking_data <- cohort_csv_data |>
2   dplyr::select(c("cohort_unique_id",
3                   "Smoke History")) |>
4   dplyr::mutate(
5     smoke_current = dplyr::case_when(
6       is.na(.data[["Smoke History"]]) ~ "-1",
7       .data[["Smoke History"]] == "non-smoker" ~ "0",
8       .data[["Smoke History"]] == "past smoker" ~ "0",
9       .data[["Smoke History"]] == "current smoker" ~ "1",
10      .default = NA_character_
11    ),
12    smoke_current = forcats::fct_relevel(
13      .data[["smoke_current"]],
14      c("0", "1")),
15    smoke_past = dplyr::case_when(
16      is.na(.data[["Smoke History"]]) ~ "-1",
17      .data[["Smoke History"]] == "non-smoker" ~ "0",
18      .data[["Smoke History"]] == "past smoker" ~ "1",
19      .data[["Smoke History"]] == "current smoker" ~ "0",
20      .default = NA_character_
21    ),
22    smoke_past = forcats::fct_relevel(
```

## 2.3 Smoking History

`smoke_current` is grouped as follows:

| Smoke History  | smoke_current |
|----------------|---------------|
| non-smoker     | 0             |
| past smoker    | 0             |
| current smoker | 1             |
| NA             | -1            |

`smoke_past` is grouped as follows:

| Smoke History  | smoke_past |
|----------------|------------|
| non-smoker     | 0          |
| past smoker    | 1          |
| current smoker | 0          |
| NA             | -1         |



# Variable Mapping

## Code to validate

```
1 smoking_data <- smoking_data |>
2   pointblank::col_vals_in_set(
3     columns = c("smoke_current", "smoke_past"),
4     set = c("0", "1", "-1")
5   ) |>
6   pointblank::col_vals_expr(
7     expr = pointblank::expr(
8       (.data[["smoke_current"]] == "1" & .data[["smoke_past"]]
9       (.data[["smoke_current"]] == "-1" & .data[["smoke_past"]]
10      (.data[["smoke_current"]] == "0" & .data[["smoke_past"]]
11    )
12  )
```

We do a check to ensure that we can only have these scenarios

- `smoke_current` as 1 and `smoke_past` as 0 for current smokers
- `smoke_current` as 0 and `smoke_past` as 1 for past smokers
- `smoke_current` as 0 and `smoke_past` as 0 for non-smokers
- `smoke_current` as -1 and `smoke_past` as -1 for unknown

Reference: <https://github.com/rstudio/pointblank/issues/578>

# Variable Mapping

Show examples that the code works

```

::: {.content-visible when-format="html"}
{r}
#| label: smoking data html
#| eval: !expr out_type == "html"

if (params$show_table && knitr::is_html_output()) {
  smoking_data |>
  harmonisation::reactable_with_download_csv_button()
}

```

Html Output

| cohort_unique_id     | Smoke History  | smoke_current | smoke_past                               |
|----------------------|----------------|---------------|--|
| <input type="text"/> | All            | All           | All                                      |
| B001                 | non-smoker     | 0             | 0  |
| B002                 | current smoker | 1             | 0  |
| B003                 | non-smoker     | 0             | 0  |
| B004                 | past smoker    | 0             | 1  |
| B005                 | current smoker | 1             | 0  |
| 1-5 of 20 rows       |                | Previous      | <input type="text" value="1"/> of 4 Next |

Download as CSV

# Variable Mapping

Show examples that the code works

```
 ::: {.content-visible unless-format="html"}
 ```{r}
 #| label: smoking data not html
 #| eval: !expr out_type != "html"

 if (params$show_table) {
   smoking_data |>
   dplyr::distinct(.data[["Smoke History"]],
   .keep_all = TRUE) |>
   knitr::kable()
 }
 ```

 :::
```

Pdf Output

```
if (params$show_table) {
  smoking_data |>
    dplyr::distinct(.data[["Smoke History"]],
                    .keep_all = TRUE) |>
    knitr::kable()
}
```

| cohort_unique_id | Smoke History  | smoke_current | smoke_past |
|------------------|----------------|---------------|------------|
| B001             | non-smoker     | 0             | 0          |
| B002             | current smoker | 1             | 0          |
| B004             | past smoker    | 0             | 1          |
| B017             | NA             | -1            | -1         |

# Variable Mapping

Clean up variables  
for merging

```
1 smoking_data <- smoking_data |>
2   dplyr::select(-c("Smoke History"))
```

| cohort_unique_id                                  | smoke_current | smoke_past |
|---|---------------|------------|
| <input type="text"/>                              | All           | All        |
| B001  | 0             | 0          |
| B002  | 1             | 0          |
| B003  | 0             | 0          |
| B004  | 0             | 1          |
| B005  | 1             | 0          |
| 1-5 of 20 rows                                    |               |            |
| Previous <input type="text" value="1"/> of 4 Next |               |            |


Download as CSV

# Merging Harmonised Data

Suppose we have completed harmonising a batch of clinical data.


```
1 age_gender_data |>
2   reactable_with_download_csv_button(
3     defaultPageSize = 5,
4     paginationType = "jump",
5     style = list(fontSize = "1rem"),
6   )
```

| cohort_unique_id     | age_years            | sex                |
|----------------------|----------------------|--------------------|
| <input type="text"/> | <input type="text"/> | All <span>▼</span> |
| B001                 | 32                   | 0                  |
| B002                 | 52                   | 0                  |
| B003                 | 80                   | 1                  |
| B004                 | 70                   | 1                  |
| B005                 | 70                   | 0                  |
| 1-5 of 20 rows       |                      |                    |
| Previous 1 of 4 Next |                      |                    |

 Download as CSV

```
1 body_measurement_data |>
2   reactable_with_download_csv_button(
3     defaultPageSize = 5,
4     paginationType = "jump",
5     style = list(fontSize = "1rem"),
6   )
```

| cohort_unique_id     | height_cm            | weight_kg            | bsa_m2               | bmi                  |
|----------------------|----------------------|----------------------|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| B001                 | 170                  | 63                   | 1.72                 | 21.8                 |
| B002                 | 167                  | 71                   | 1.81                 | 25.46                |
| B003                 | 184                  | 77                   | 1.98                 | 22.74                |
| B004                 | 160                  | 83                   | 1.92                 | 32.42                |
| B005                 | 155                  | 61                   | 1.62                 | 25.39                |
| 1-5 of 20 rows       |                      |                      |                      |                      |
| Previous 1 of 4 Next |                      |                      |                      |                      |

 Download as CSV

How can we merge them without issues of missing rows or additional columns ?



# Merging Harmonised Data

`unmatched = "error"` in `dplyr::inner_join` helps to avoid patients with no match.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behave_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6                     by = join_specification,
7                     unmatched = "error",
8                     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10                    by = join_specification,
11                    unmatched = "error",
12                    relationship = "one-to-one") |>
13   dplyr::inner_join(smoking_data,
14                    by = join_specification,
15                    unmatched = "error",
16                    relationship = "one-to-one") |>
17   dplyr::relocate(c("bsa_m2", "bmi"),
18                  .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species,   ~island,
3   1,        "Adelie",   "Torgersen",
4   2,        "Gentoo",   "Biscoe",
5 )
6
7 weight_extra <- tibble::tribble(
8   ~samp_id, ~body_mass_g,
9   1,        3220,
10  2,        4730,
11  4,        4725
12 )
13
14 three_penguins |>
15   dplyr::inner_join(
16     y = weight_extra,
17     by = dplyr::join_by("samp_id"),
18     unmatched = "error"
19   )
```

```
Error in `dplyr::inner_join()`:
! Each row of `y` must be matched by `x`.
i Row 3 of `y` was not matched.
```

Reference: <https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-join-functionality>

# Merging Harmonised Data

`unmatched = "error"` in `dplyr::inner_join` helps to avoid patients with no match.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behave_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6                     by = join_specification,
7                     unmatched = "error",
8                     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10                    by = join_specification,
11                    unmatched = "error",
12                    relationship = "one-to-one") |>
13   dplyr::inner_join(smoking_data,
14                    by = join_specification,
15                    unmatched = "error",
16                    relationship = "one-to-one") |>
17   dplyr::relocate(c("bsa_m2", "bmi"),
18                  .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1,        "Adelie",  "Torgersen",
4   2,        "Gentoo",  "Biscoe",
5   3,        "Chinstrap", "Dream"
6 )
7
8 weight_extra <- tibble::tribble(
9   ~samp_id, ~body_mass_g,
10  1,        3220,
11  3,        4725
12 )
13
14 three_penguins |>
15   dplyr::inner_join(
16     y = weight_extra,
17     by = dplyr::join_by("samp_id"),
18     unmatched = "error"
19   )
```

```
Error in `dplyr::inner_join()`:
! Each row of `x` must have a match in `y`.
i Row 2 of `x` does not have a match.
```

Reference: <https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-join-functionality>

# Merging Harmonised Data

`relationship = "one-to-one"` in `dplyr::inner_join` helps to avoid patients with multiple match.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behave_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6                     by = join_specification,
7                     unmatched = "error",
8                     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10                    by = join_specification,
11                    unmatched = "error",
12                    relationship = "one-to-one") |>
13   dplyr::inner_join(smoking_data,
14                    by = join_specification,
15                    unmatched = "error",
16                    relationship = "one-to-one") |>
17   dplyr::relocate(c("bsa_m2", "bmi"),
18                  .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1,        "Adelie",  "Torgersen",
4   2,        "Gentoo",  "Biscoe",
5   3,        "Chinstrap", "Dream"
6 )
7
8 weight_extra <- tibble::tribble(
9   ~samp_id, ~body_mass_g,
10  1,        3220,
11  2,        4730,
12  2,        4725,
13  3,        4000
14 )
15
16 three_penguins |>
17   dplyr::inner_join(
18     y = weight_extra,
19     by = dplyr::join_by("samp_id"),
20     relationship = "one-to-one"
21   )
```

```
Error in `dplyr::inner_join()`:  
! Each row in `x` must match at most 1 row in `y`.  
i Row 2 of `x` matches multiple rows in `y`.
```

Reference: <https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-join-functionality>

# Merging Harmonised Data

Use `pointblank::has_columns` to ensure we only have harmonised variables.

```
1 testthat::expect_false(  
2   pointblank::has_columns(  
3     demo_behave_data,  
4     columns = c(  
5       dplyr::ends_with(".x"),  
6       dplyr::ends_with(".y")  
7     )  
8   )  
9 )  
10  
11 testthat::expect_equal(  
12   ncol(demo_behave_data), 9  
13 )  
14  
15 testthat::expect_true(  
16   pointblank::has_columns(  
17     demo_behave_data,  
18     columns = c(  
19       "age_years", "sex",  
20       "height_cm", "weight_kg", "bsa_m2", "bmi",  
21       "smoke_current", "smoke_past"  
22     )  
23   )  
24 )
```

```
1 three_penguins <- tibble::tribble(  
2   ~samp_id, ~species, ~island,  
3   1,        "Adelie",  "Torgersen",  
4   2,        "Gentoo",  "Biscoe",  
5   3,        "Chinstrap", "Dream"  
6 )  
7  
8 weight_extra <- tibble::tribble(  
9   ~samp_id, ~island,  
10  1,        "Torgersen",  
11  2,        "Biscoe",  
12  3,        "Dream"  
13 )  
14  
15 three_penguins <- three_penguins |>  
16   dplyr::inner_join(  
17     y = weight_extra,  
18     by = dplyr::join_by("samp_id"),  
19     unmatched = "error",  
20     relationship = "one-to-one"  
21   )  
22
```

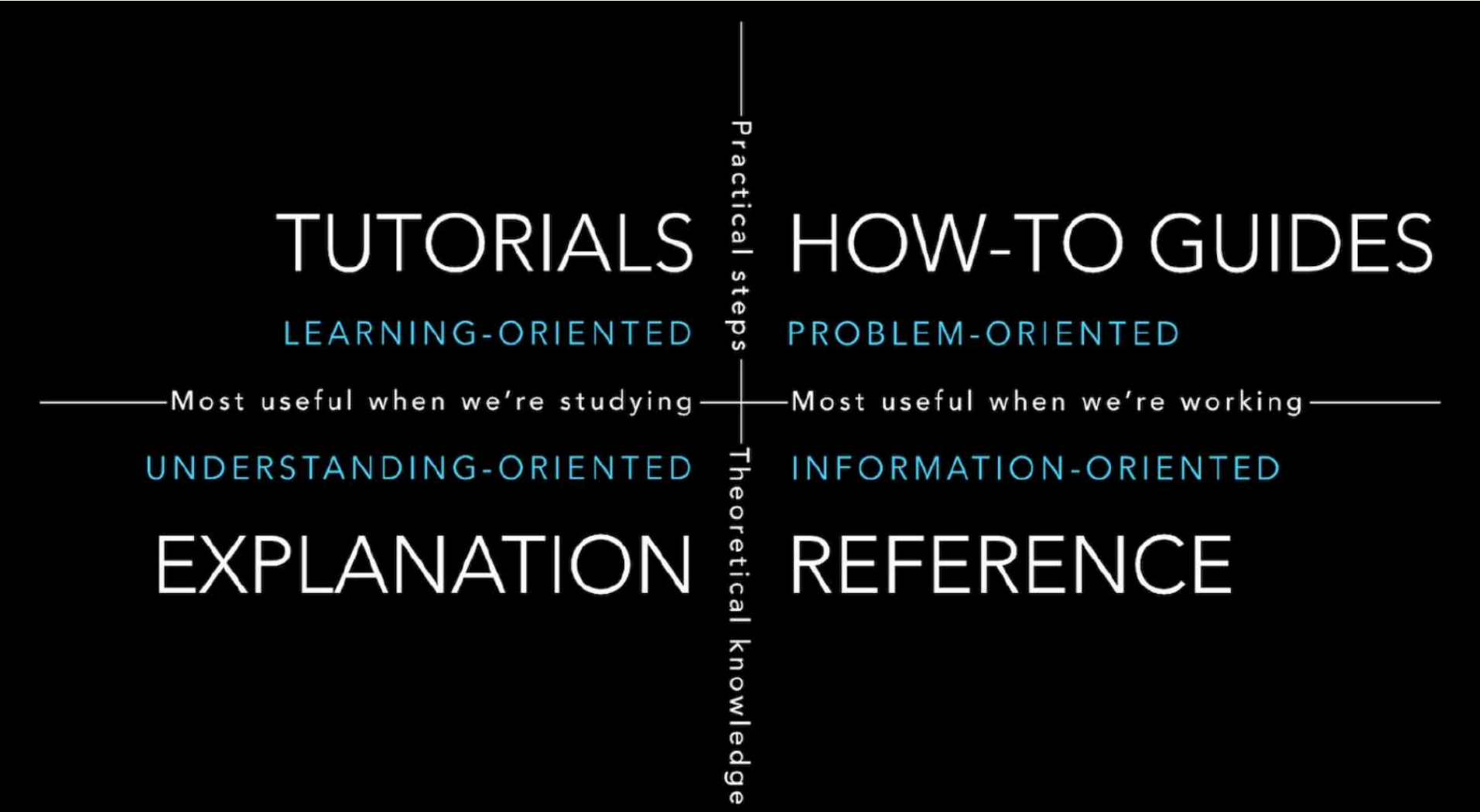
```
[1] TRUE
```

```
1 colnames(three_penguins)
```

```
[1] "samp_id" "species" "island.x" "island.y"
```

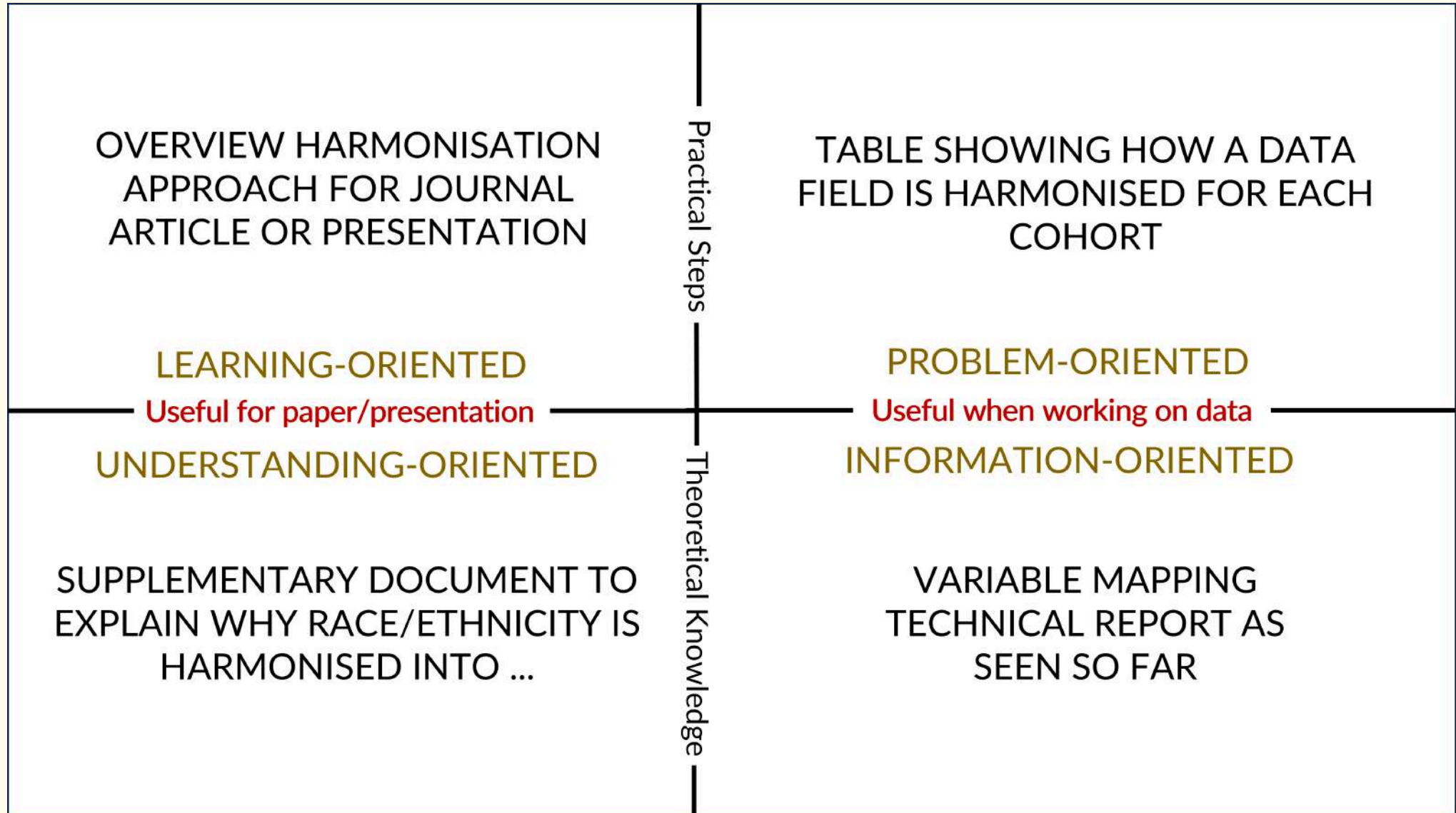
# Harmonisation Report Types

Collaborator wants different ways to report how data harmonisation is done.



# Harmonisation Report Types

Collaborator wants different ways to report how data harmonisation is done.





# Tehcnical Report Challenge

One variable mapping report takes at least one page.

On average, a clinical trial will have a few hundred variables.

- One hundred columns for clinical and demographics.
- Two hundred columns for medication.

Harmonisation report can have at least a few hundreds pages for each cohort.

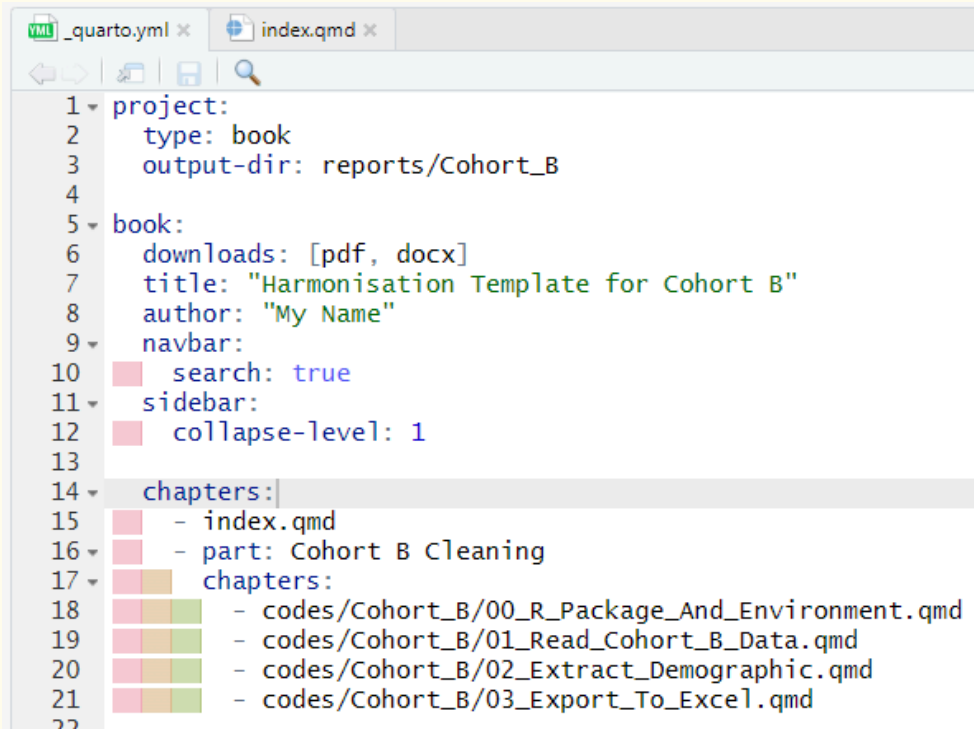
There is a need to automate the creation of these reports.



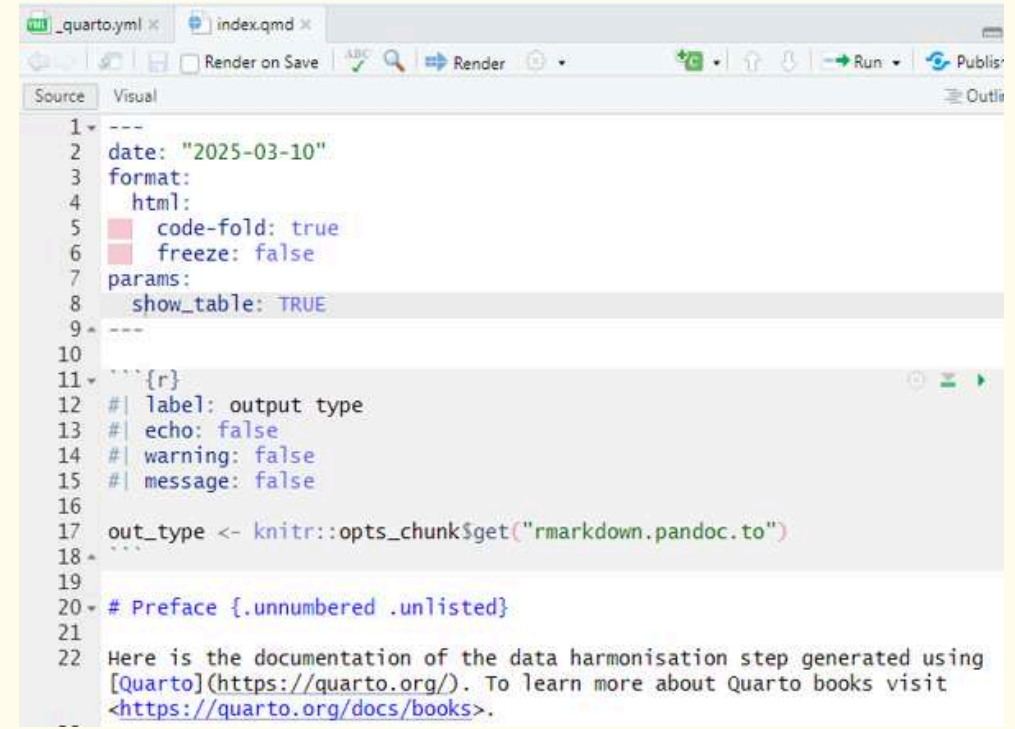
Businessman in pile of documents asking for help by [Amonrat Rungreangfangsai](#)

# Quarto Books

To make a Quarto book or website, we need a `_quarto.yml` and `index.qmd` file



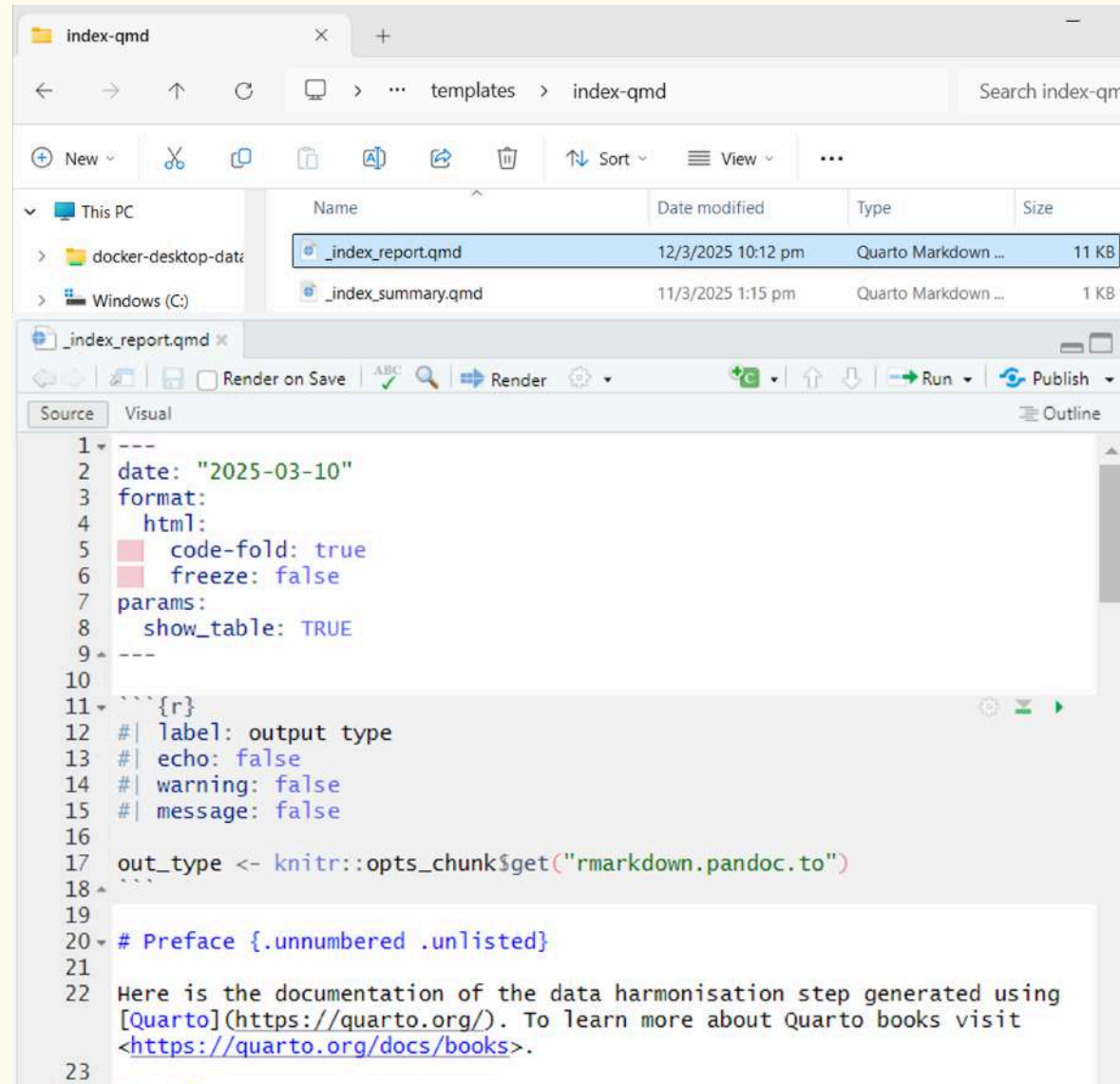
```
1 project:
2   type: book
3   output-dir: reports/Cohort_B
4
5 book:
6   downloads: [pdf, docx]
7   title: "Harmonisation Template for Cohort B"
8   author: "My Name"
9   navbar:
10    search: true
11   sidebar:
12    collapse-level: 1
13
14 chapters:
15   - index.qmd
16   - part: Cohort B Cleaning
17     chapters:
18       - codes/Cohort_B/00_R_Package_And_Environment.qmd
19       - codes/Cohort_B/01_Read_Cohort_B_Data.qmd
20       - codes/Cohort_B/02_Extract_Demographic.qmd
21       - codes/Cohort_B/03_Export_To_Excel.qmd
22
```



```
1 ---
2 date: "2025-03-10"
3 format:
4   html:
5     code-fold: true
6     freeze: false
7 params:
8   show_table: TRUE
9 ---
10
11 {r}
12 #| label: output type
13 #| echo: false
14 #| warning: false
15 #| message: false
16
17 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
18 ...
19
20 # Preface {.unnumbered .unlisted}
21
22 Here is the documentation of the data harmonisation step generated using
23 [Quarto](https://quarto.org/). To learn more about Quarto books visit
24 <https://quarto.org/docs/books>.
```

# Automated Technical Report (Reference)

We create an `index.qmd` file for technical report generation.

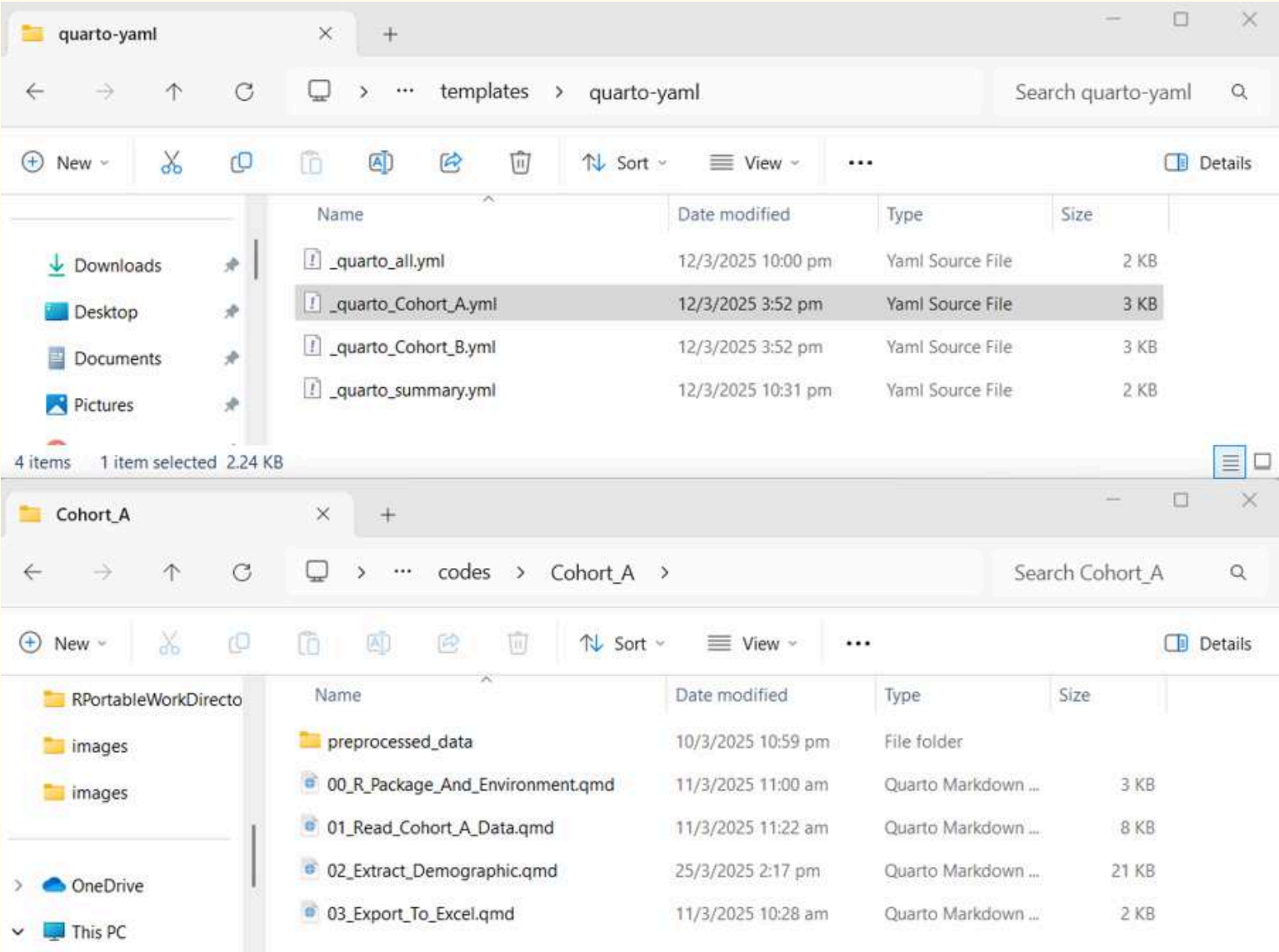


The screenshot shows a file explorer window with the path `templates > index-qmd`. It contains two files: `_index_report.qmd` (11 KB, modified 12/3/2025 10:12 pm) and `_index_summary.qmd` (1 KB, modified 11/3/2025 1:15 pm). Below the file explorer is a code editor window showing the content of `_index_report.qmd`. The code is as follows:

```
1 ---
2 date: "2025-03-10"
3 format:
4   html:
5     code-fold: true
6     freeze: false
7 params:
8   show_table: TRUE
9 ---
10
11 {r}
12 #| label: output type
13 #| echo: false
14 #| warning: false
15 #| message: false
16
17 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
18
19
20 # Preface {.unnumbered .unlisted}
21
22 Here is the documentation of the data harmonisation step generated using
23 [Quarto](https://quarto.org/). To learn more about Quarto books visit
24 <https://quarto.org/docs/books>.
```

# Automated Technical Report (Reference)

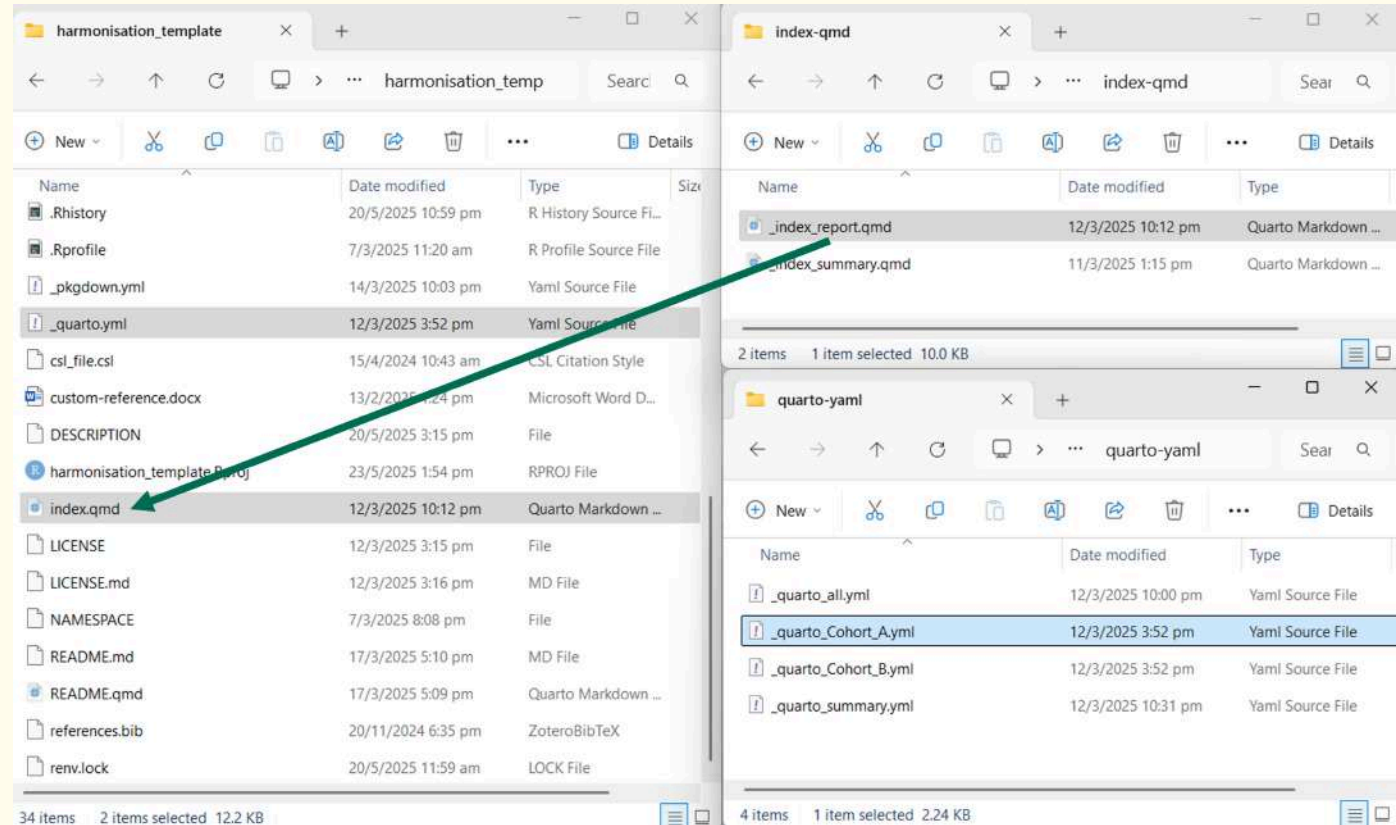
We create a `_quarto.yml` file and relevant Quarto files for each cohort.



# Automated Technical Report (Reference)

Create a script to generate technical reports in pdf, word and html for each cohort.

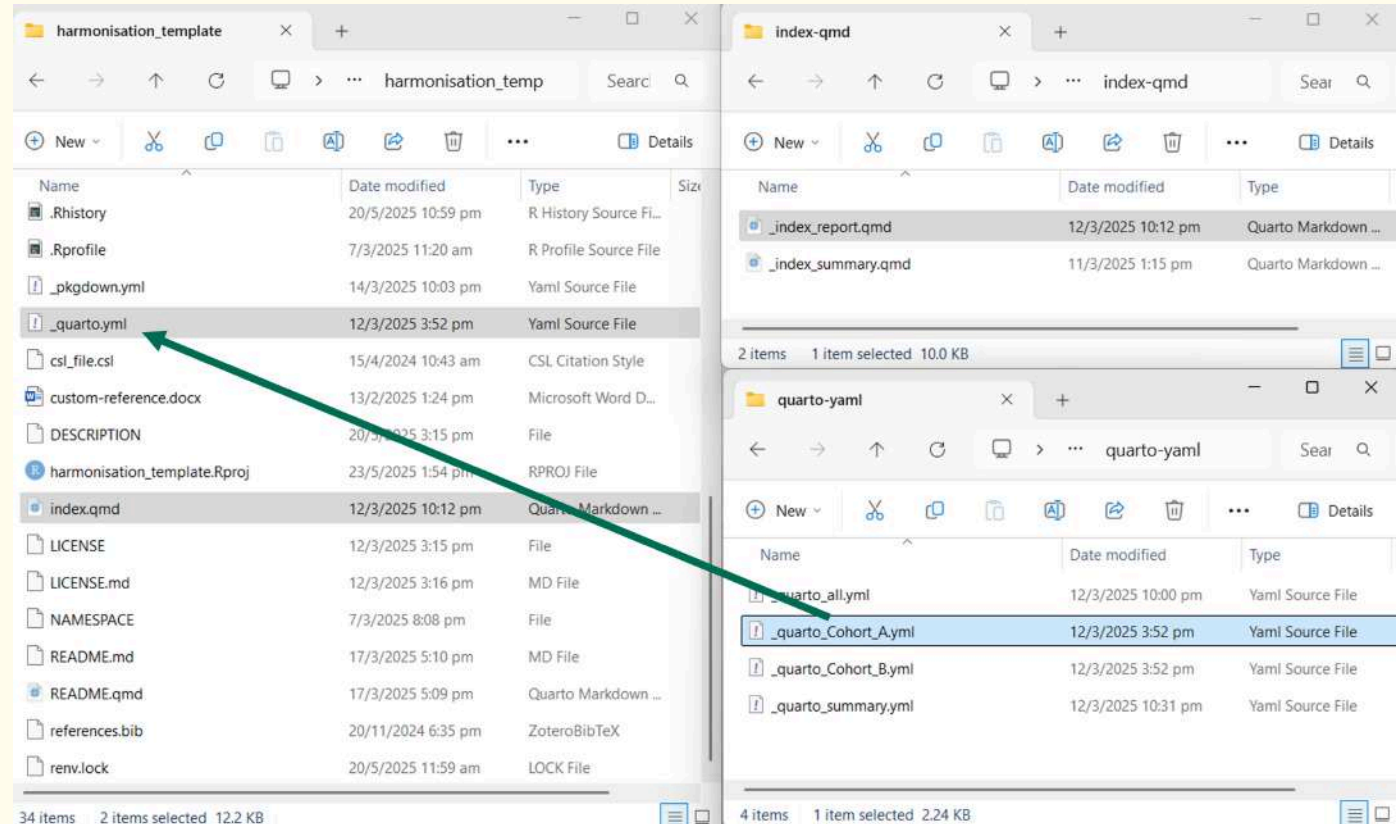
```
1 # Copy the right index.qmd
2 # file
3
4 index_qmd_file <- paste0(
5   "_index_",
6   "report",
7   ".qmd"
8 )
9
10 fs::file_copy(
11   path = here::here(
12     "templates",
13     "index-qmd",
14     index_qmd_file),
15   new_path = here::here(
16     "index.qmd"
17   ),
18   overwrite = TRUE
19 )
```



# Automated Technical Report (Reference)

Create a script to generate technical reports in pdf, word and html for each cohort.

```
1 copy_and_render <- function(  
2   cohort  
3 ) {  
4  
5   # Copy quarto.yml file  
6   # for each cohort  
7  
8   quarto_yml_file <- paste0(  
9     "_quarto_",  
10    cohort,  
11    ".yml"  
12  )  
13  
14  fs::file_copy(  
15    path = here::here(  
16      "templates",  
17      "quarto-yaml",  
18      quarto_yml_file),  
19    new_path = here::here("_quarto",  
20      overwrite = TRUE  
21    )  
22  )
```



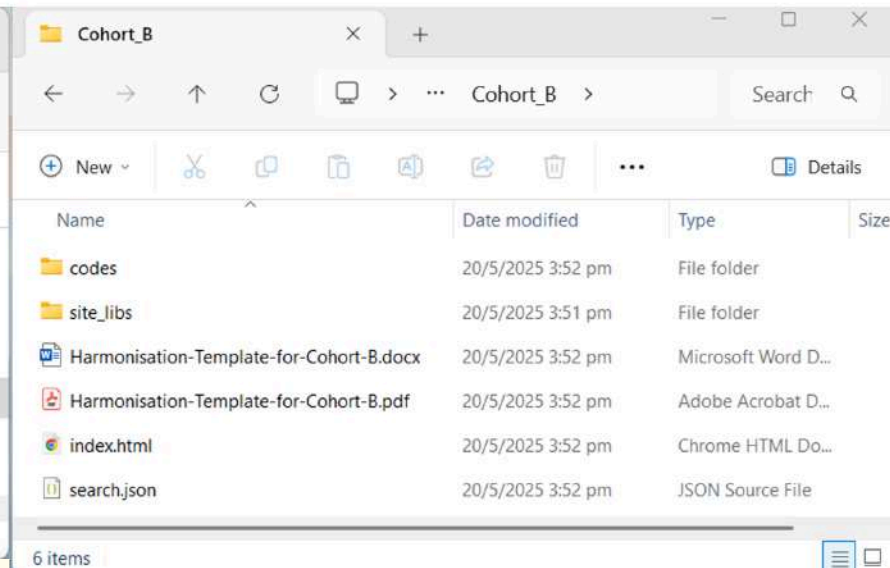
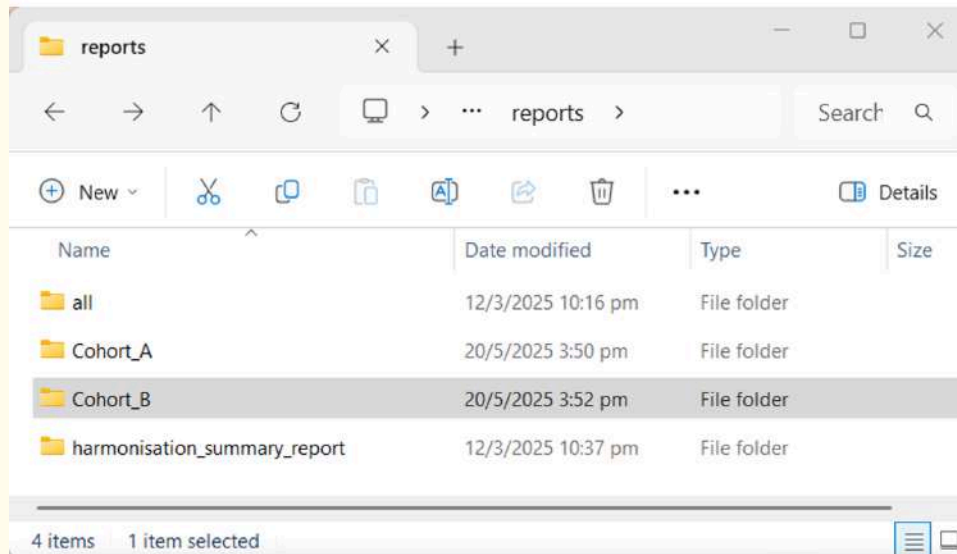


# Automated Technical Report (Reference)

Output of these reports are as follows:

Run the R script `cohort_harmonisation_script.R` in `codes` folder to generate:

- Cohort\_A Harmonisation Report:
  -  HTML: <https://jauntyjjs-harmonisation-cohort-a.netlify.app>
  -  PDF : <https://jauntyjjs-harmonisation-cohort-a.netlify.app/Harmonisation-Template-for-Cohort-A.pdf>
  -  Word: <https://jauntyjjs-harmonisation-cohort-a.netlify.app/Harmonisation-Template-for-Cohort-A.docx>
- Cohort\_B Harmonisation Report:
  -  HTML: <https://jauntyjjs-harmonisation-cohort-b.netlify.app>
  -  PDF : <https://jauntyjjs-harmonisation-cohort-b.netlify.app/Harmonisation-Template-for-Cohort-B.pdf>
  -  Word: <https://jauntyjjs-harmonisation-cohort-b.netlify.app/Harmonisation-Template-for-Cohort-B.docx>



# Automated Summary Report (How-to-Guide)

A similar method is done to create a summary report in word using [flextable](#).

## 2.4 Smoking History

*smoke\_current* is the harmonised data field to denote if the patient is a current smoker during the time of the CT scan. *smoke\_past* is the harmonised data field to denote if the patient is a past smoker during the time of the CT scan.

They hold the following values:

Table S6: Harmonised values of *smoke\_current* and *smoke\_past*.

| Value | Description |
|-------|-------------|
| 0     | no          |
| 1     | yes         |
| -1    | unknown     |

They are harmonised as follows:

Table S7: Harmonised process of *smoke\_current* and *smoke\_past*.

| Cohort ID | Original Response  | Harmonisation Response   |
|-----------|--|--|
| Cohort A  | Column <i>smoke_current_good</i> with                                | <i>smoke_current</i> will take the values of <i>smoke_current_good</i> .<br><br><i>smoke_past</i> will take the values of <i>smoke_past_good</i> . |
|           | 0 as no.   |  |
|           | 1 as yes.  |  |
|           | -1 as unknown.   |  |
|           | Column <i>smoke_past_good</i> with                                   |  |
|           | 0 as no.   |  |
| Cohort B  | 1 as yes.  | Map the values of <i>Smoke History</i> to <i>smoke_current</i> as follows:   |
|           | -1 as unknown.   |  |
|           | Column <i>Smoke History</i> with<br><i>non-smoker</i> as non-smoker. |  |

|  |   |
|--|---|
| <i>past smoker</i> as a past smoker.       | <i>non-smoker</i> and <i>past smoker</i> as 0.                          |
| <i>current smoker</i> as a current smoker. | <i>current smoker</i> as 1.   |
| NA as unknown.                             | NA as -1.   |
|  | Map the values of <i>Smoke History</i> to <i>smoke_past</i> as follows: |
|  | <i>non-smoker</i> and <i>current smoker</i> as 0.                       |
|  | <i>past smoker</i> as 1.  |
|  | NA as -1.   |

After harmonisation, we validate the values of *smoke\_current* and *smoke\_past* to ensure that there can only be the following cases:

Table S8: Valid values of *smoke\_current* and *smoke\_past*.

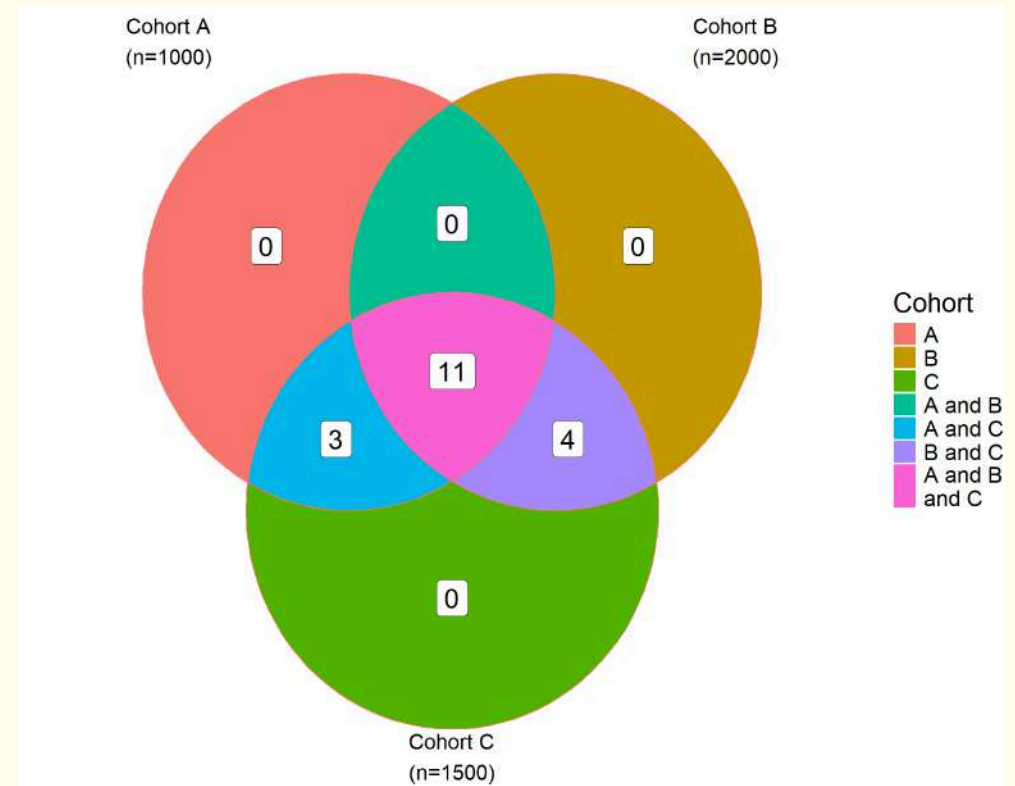
| Description    | <i>smoke_current</i> | <i>smoke_past</i> |
|----------------|----------------------|-------------------|
| Non-smoker     | 0                    | 0                 |
| Past smoker    | 0                    | 1                 |
| Current smoker | 1                    | 0                 |
| Unknown        | -1                   | -1                |

# Overview Diagrams

How many variables can each cohort provide ?

How many variables can be harmonised ?

```
1 demographic_list <- list(  
2   A = c("Age", "Sex",  
3       "Hypertension", "Dyslipidemia", "Family Hx CAD", "Diabe  
4       "Smoke Current", "Smoke Past",  
5       "Have Chest Pain", "Chest Pain Character",  
6       "Dyspnea",  
7       "BMI", "Height", "Weight"),  
8   B = c("Age", "Sex",  
9       "Hypertension", "Dyslipidemia", "Family Hx CAD", "Diabe  
10      "Smoke Current", "Smoke Past",  
11      "Have Chest Pain", "Chest Pain Character",  
12      "Dyspnea",  
13      "HDL", "Total Cholesterol",  
14      "Triglyceride", "LDL"),  
15   C = c("Age", "Sex",  
16       "Hypertension", "Dyslipidemia", "Family Hx CAD", "Diabe  
17       "Smoke Current", "Smoke Past",  
18       "Have Chest Pain", "Chest Pain Character",  
19       "Dyspnea",  
20       "BMI", "Height", "Weight",  
21       "HDL", "Total Cholesterol",  
22       "Triglyceride", "LDL")
```

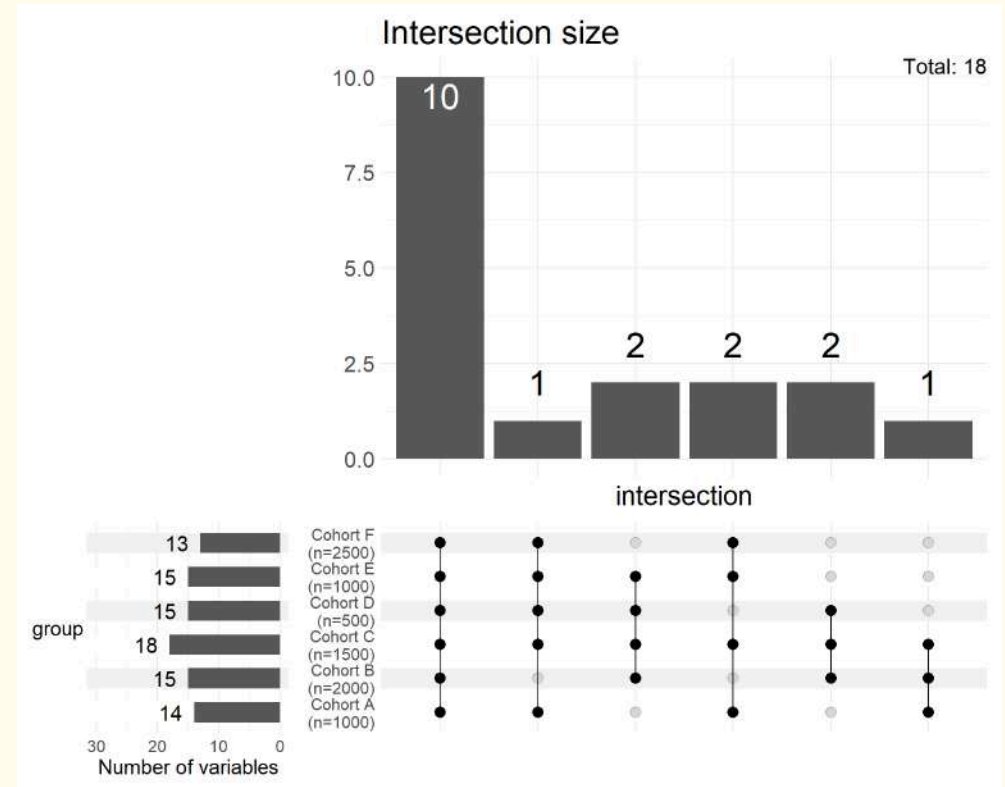


Venn diagram does not work for many (> 10) cohorts.

# Overview Diagrams

Upset plots are too complicated for clinicians.

```
1 demographic_venn <- tibble::tibble(  
2   column_name = c("Age", "Sex",  
3                   "Hypertension", "Dyslipidemia", "Family Hx CA",  
4                   "Smoke Current", "Smoke Past",  
5                   "Have Chest Pain", "Chest Pain Character",  
6                   "Dyspnea",  
7                   "BMI", "Height", "Weight",  
8                   "HDL", "Total Cholesterol",  
9                   "Triglyceride", "LDL"),  
10  `Cohort A` = c(1, 1,  
11                1, 1, 1, 1,  
12                1, 1,  
13                1, 1,  
14                1,  
15                1, 1, 1,  
16                0, 0,  
17                0, 0),  
18  `Cohort B` = c(1, 1,  
19                1, 1, 1, 1,  
20                1, 1,  
21                1, 1,  
22                1,
```

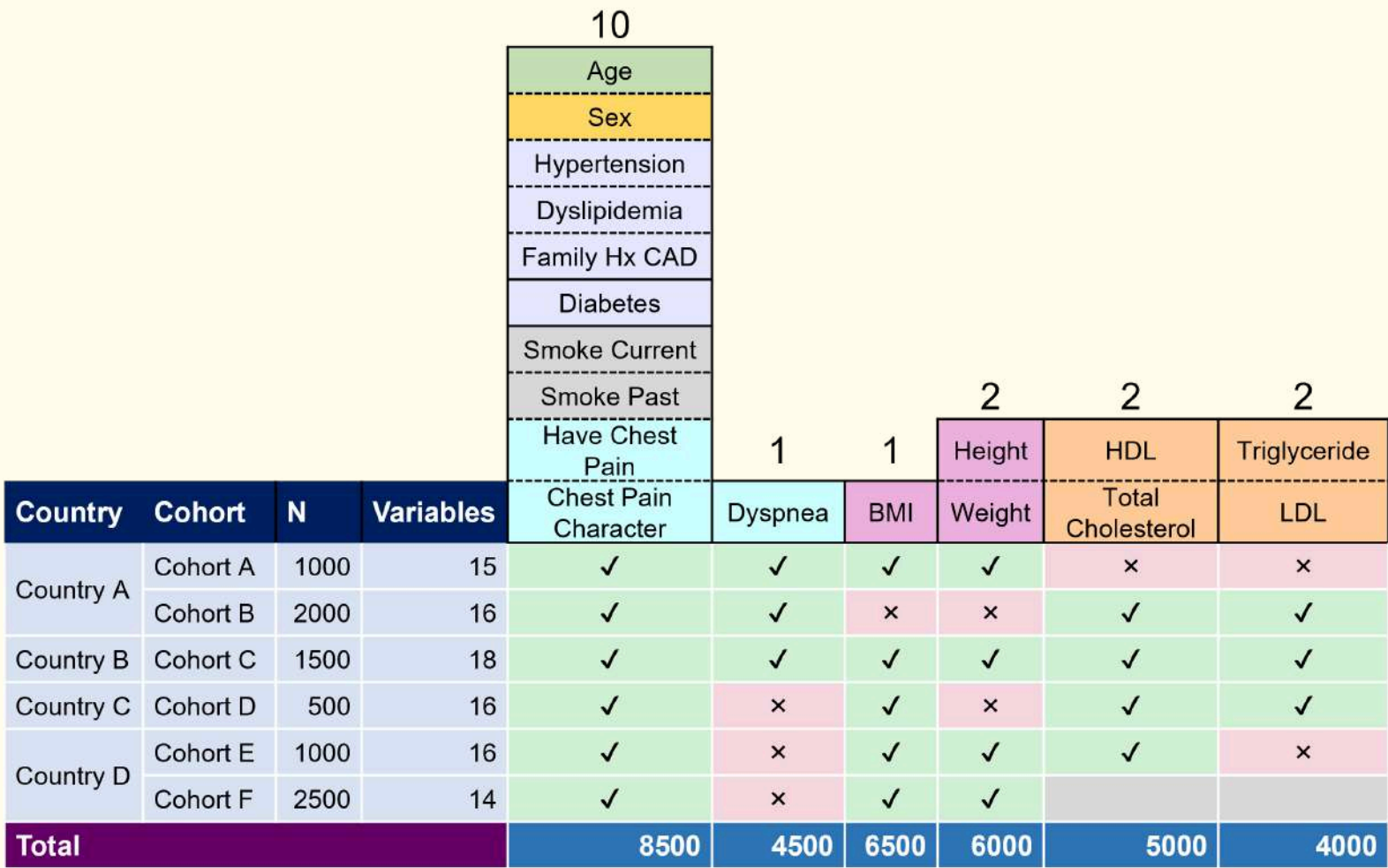


Cannot answer follow-up questions:

How many cohorts provide patient's blood lipid information and how many patients have them ?

# Overview Diagrams

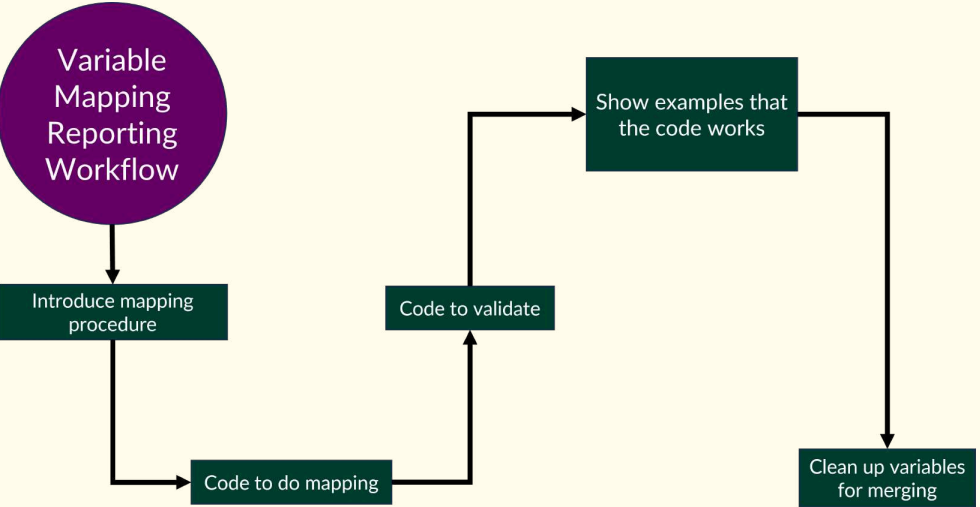
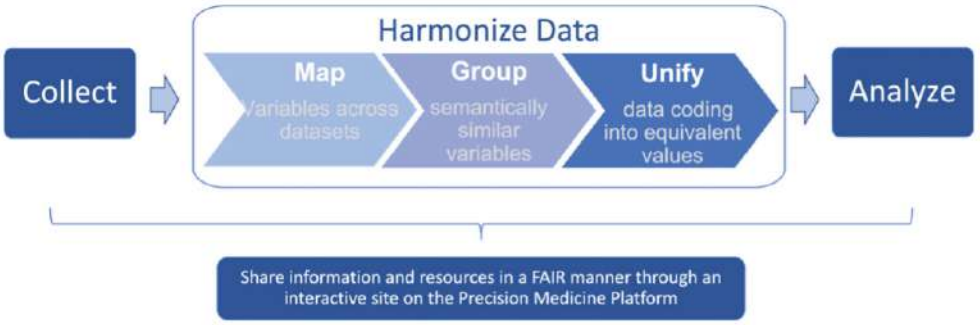
Create a “heatmap”using Microsoft PowerPoint.



| Variable Colour Legend |                 |
|------------------------|-----------------|
|                        | Age             |
|                        | Sex             |
|                        | Comorbidity     |
|                        | Smoking history |
|                        | Symptoms        |
|                        | Obesity         |
|                        | Blood lipid     |

| Table Legend |                 |
|--------------|-----------------|
| ✓            | Available       |
| ×            | Not available   |
|              | Pending arrival |

# Summary



|   |                       |  |
|---|-----------------------|--|
| OVERVIEW HARMONISATION APPROACH FOR JOURNAL ARTICLE OR PRESENTATION         | Practical Steps       | TABLE SHOWING HOW A DATA FIELD IS HARMONISED FOR EACH COHORT |
| LEARNING-ORIENTED<br>Useful for paper/presentation                          |                       | PROBLEM-ORIENTED<br>Useful when working on data              |
| UNDERSTANDING-ORIENTED  | Theoretical Knowledge | INFORMATION-ORIENTED   |
| SUPPLEMENTARY DOCUMENT TO EXPLAIN WHY RACE/ETHNICITY IS HARMONISED INTO ... |                       | VARIABLE MAPPING TECHNICAL REPORT AS SEEN SO FAR             |

|           |          |                      |           |      |  |      |  |        |  |                   |  |
|-----------|----------|----------------------|-----------|------|--|------|--|--------|--|-------------------|--|
|           |          | 10                   |           |      |  |      |  |        |  |                   |  |
|           |          | Age                  |           |      |  |      |  |        |  |                   |  |
|           |          | Sex                  |           |      |  |      |  |        |  |                   |  |
|           |          | Hypertension         |           |      |  |      |  |        |  |                   |  |
|           |          | Dyslipidemia         |           |      |  |      |  |        |  |                   |  |
|           |          | Family Hx CAD        |           |      |  |      |  |        |  |                   |  |
|           |          | Diabetes             |           |      |  |      |  |        |  |                   |  |
|           |          | Smoke Current        |           |      |  |      |  |        |  |                   |  |
|           |          | Smoke Past           |           |      |  |      |  |        |  |                   |  |
|           |          | Have Chest Pain      |           |      |  |      |  |        |  |                   |  |
|           |          | Chest Pain Character |           | 1    |  | 1    |  | 2      |  | 2                 |  |
|           |          | Dyspnea              |           |      |  |      |  | Height |  | HDL               |  |
|           |          | BMI                  |           |      |  |      |  | Weight |  | Total Cholesterol |  |
|           |          |                      |           |      |  |      |  |        |  | Triglyceride      |  |
| Country   | Cohort   | N                    | Variables |      |  |      |  |        |  |                   |  |
| Country A | Cohort A | 1000                 | 15        | ✓    |  | ✓    |  | ✓      |  | x                 |  |
|           | Cohort B | 2000                 | 16        | ✓    |  | ✓    |  | x      |  | ✓                 |  |
| Country B | Cohort C | 1500                 | 18        | ✓    |  | ✓    |  | ✓      |  | ✓                 |  |
|           | Cohort D | 500                  | 16        | ✓    |  | x    |  | x      |  | ✓                 |  |
| Country C | Cohort E | 1000                 | 16        | ✓    |  | x    |  | ✓      |  | ✓                 |  |
|           | Cohort F | 2500                 | 14        | ✓    |  | x    |  | ✓      |  | x                 |  |
| Total     |          |                      |           | 8500 |  | 4500 |  | 6500   |  | 5000              |  |

Variable Colour Legend

|                 |
|-----------------|
| Age             |
| Sex             |
| Comorbidity     |
| Smoking history |
| Symptoms        |
| Obesity         |
| Blood lipid     |

Table Legend

|   |                 |
|---|-----------------|
| ✓ | Available       |
| x | Not available   |
|   | Pending arrival |



# Thank you

Harmonisation project template: <https://github.com/JauntyJJS/harmonisation/>

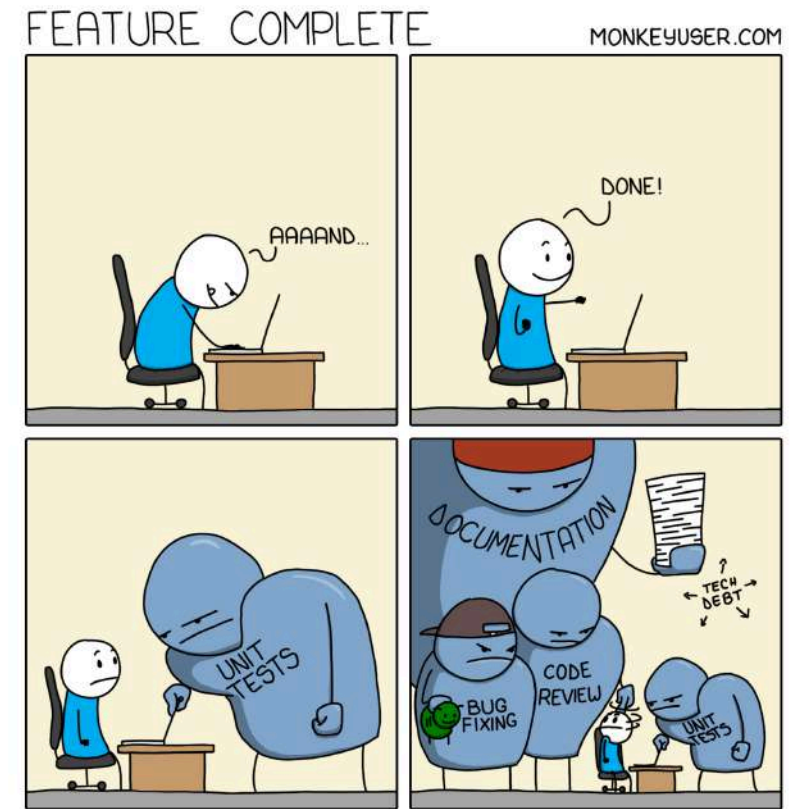
README License MIT license

## Data Harmonisation Project Template

R-CMD-check.yaml passing

### Table of Content

- [Motivation](#)
- [Acknowledgement](#)
- [File Structure](#)
- [Software Installation](#)
- [R Package Installation](#)
- [Using `renv`](#)
- [R Functions Management](#)
- [R Packages Used](#)
- [R Platform Information](#)
- [Data Harmonisation Report For Each Cohort](#)
- [Combined Data Harmonisation Report For All Cohort](#)
- [Data Harmonisation Summary](#)
- [General Recommendations](#)



Feature Complete from MonkeyUser.com

# Happy 60th Birthday Singapore

